



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

*G10L 19/00* (2015.12); *G10L 19/02* (2015.12); *G10L 19/002* (2015.12); *G10L 19/0208* (2015.12); *G10L 19/032* (2015.12); *G10L 19/06* (2015.12)

(21)(22) Заявка: 2012141243, 11.01.2011

(24) Дата начала отсчета срока действия патента:  
11.01.2011Дата регистрации:  
07.02.2018

Приоритет(ы):

(30) Конвенционный приоритет:  
12.01.2010 US 61/294,357

(43) Дата публикации заявки: 10.08.2015 Бюл. № 22

(45) Опубликовано: 07.02.2018 Бюл. № 4

(85) Дата начала рассмотрения заявки РСТ на  
национальной фазе: 23.11.2012(86) Заявка РСТ:  
EP 2011/050273 (11.01.2011)(87) Публикация заявки РСТ:  
WO 2011/086066 (21.07.2011)

Адрес для переписки:

410000, г. Саратов, главпочтамт, а/я 62, ООО  
"ПатентВолгаСервис", Романовой Н.В.

(72) Автор(ы):

ФУШ Гильом (DE),  
МУЛТРУС Маркус (DE),  
РЕТТЕЛБАХ Николаус (DE),  
СУББАРАМАН Вигнеш (DE),  
ВЕИС Оливер (DE),  
ГАЙЕР Марк (DE),  
ВАРМБОЛД Патрик (DE),  
ГРИЕБЕЛ Кристиан (DE)

(73) Патентообладатель(и):

Фраунхофер-Гезелльшафт цур Фёрдерунг  
дер ангевандтен Форшунг Е.Ф., (DE)

(56) Список документов, цитированных в отчете  
о поиске: US 2005/0192799 A1, 01.09.2005. US  
2009/0299756 A1, 03.12.2009. US 2009/0192791  
A1, 30.07.2009. US 2006/0235679 A1, 19.10.2009.  
US 2004/0184537 A1, 23.09.2004. RU 2335809  
C2, 10.10.2008.

(54) АУДИО КОДЕР, АУДИО ДЕКОДЕР, СПОСОБ КОДИРОВАНИЯ АУДИО ИНФОРМАЦИИ,  
СПОСОБ ДЕКОДИРОВАНИЯ АУДИО ИНФОРМАЦИИ И КОМПЬЮТЕРНАЯ ПРОГРАММА,  
ИСПОЛЬЗУЮЩАЯ МОДИФИКАЦИЮ ЧИСЛОВОГО ПРЕДСТАВЛЕНИЯ ЧИСЛОВОГО ЗНАЧЕНИЯ  
ПРЕДЫДУЩЕГО КОНТЕКСТА

(57) Реферат:

Изобретение относится к области кодирования/декодирования аудио информации. Технический результат – обеспечение улучшенного компромисса между эффективностью скорости передачи битов и эффективностью использования ресурсов. Аудио декодер включает: арифметический декодер для обеспечения множества декодированных спектральных значений на основе арифметически

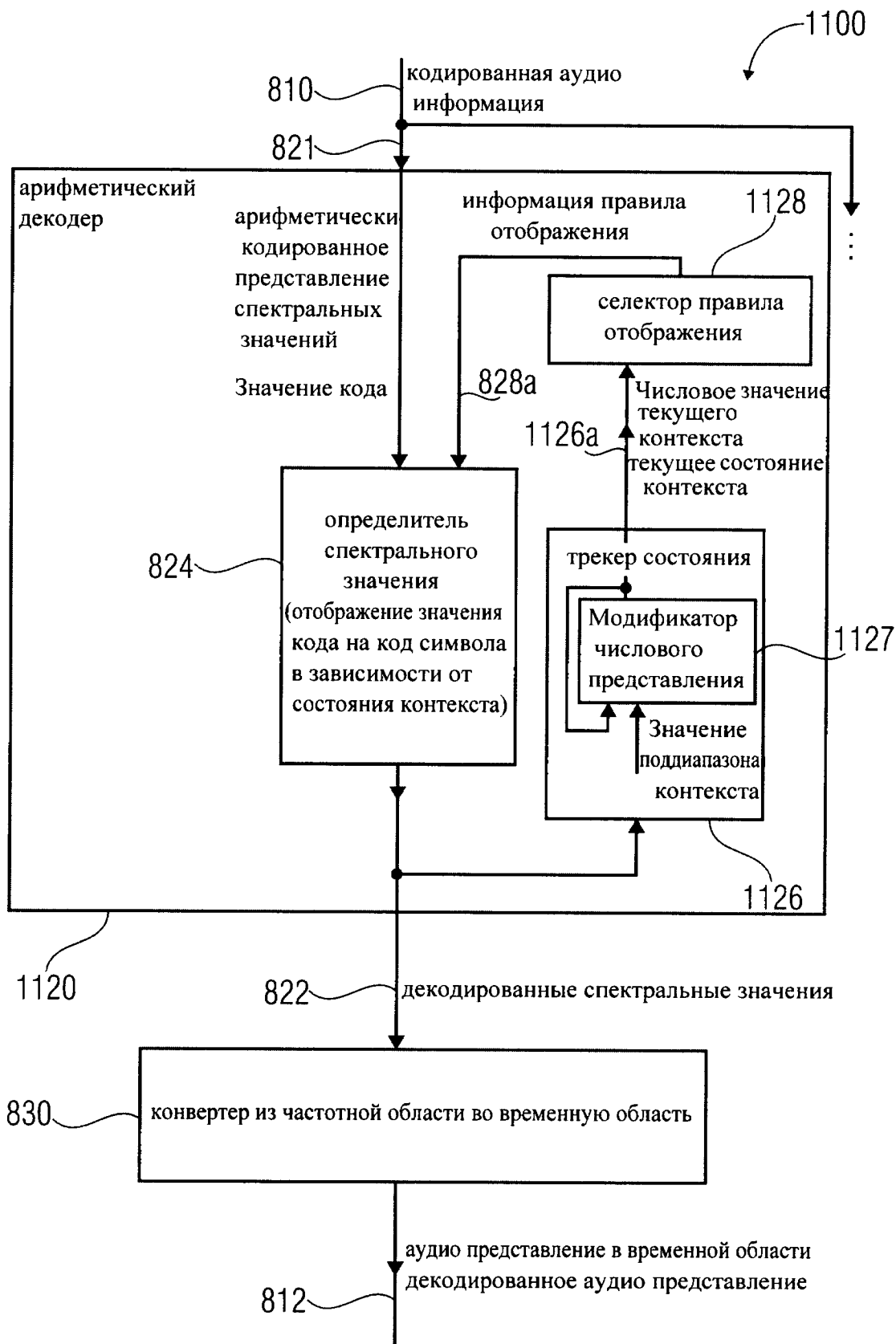
кодированного представления спектральных значений, содержащихся в кодированной аудио информации, и конвертер частотной области во временную область для обеспечения аудио представления временной области, используя декодированные спектральные значения, в целях получения декодированной аудио информации; при этом арифметический декодер настроен выбирать правило отображения, описывающее

RU 2 644 141 C2

RU 2 644 141 C2

отображение значения кода арифметически кодированного представления спектральных значений на код символа; определять числовое значение текущего контекста; и модифицировать числовое представление числового значения

предыдущего контекста в зависимости от значения поддиапазона контекста, чтобы получить числовое представление числового значения текущего контекста. 6 н. и 14 з.п. ф-лы, 67 ил.



ФИГ. 11



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

*G10L 19/00* (2015.12); *G10L 19/02* (2015.12); *G10L 19/002* (2015.12); *G10L 19/0208* (2015.12); *G10L 19/032* (2015.12); *G10L 19/06* (2015.12)

(21)(22) Application: **2012141243, 11.01.2011**(24) Effective date for property rights:  
**11.01.2011**Registration date:  
**07.02.2018**

Priority:

(30) Convention priority:  
**12.01.2010 US 61/294,357**(43) Application published: **10.08.2015 Bull. № 22**(45) Date of publication: **07.02.2018 Bull. № 4**(85) Commencement of national phase: **23.11.2012**(86) PCT application:  
**EP 2011/050273 (11.01.2011)**(87) PCT publication:  
**WO 2011/086066 (21.07.2011)**Mail address:  
**410000, g. Saratov, glavpochtamt, a/ya 62, OOO  
"PatentVolgaServis", Romanovoj N.V.**

(72) Inventor(s):

**FUSH Gilom (DE),  
MULTRUS Markus (DE),  
RETTELBAKH Nikolaus (DE),  
SUBBARAMAN Vignesh (DE),  
VEIS Oliver (DE),  
GAJER Mark (DE),  
VARMBOLD Patrik (DE),  
GRIEBEL Kristian (DE)**

(73) Proprietor(s):

**Fraunhofer-Gezellschaft tsur Ferderung der  
angevandten Forshung E.F., (DE)**

(54) **AUDIO CODER, AUDIO DECODER, AUDIO INFORMATION CODING METHOD, AUDIO INFORMATION DECODING METHOD, AND COMPUTER PROGRAM USING MODIFICATION OF NUMERICAL REPRESENTATION OF PREVIOUS CONTEXT NUMERICAL VALUE**

(57) Abstract:

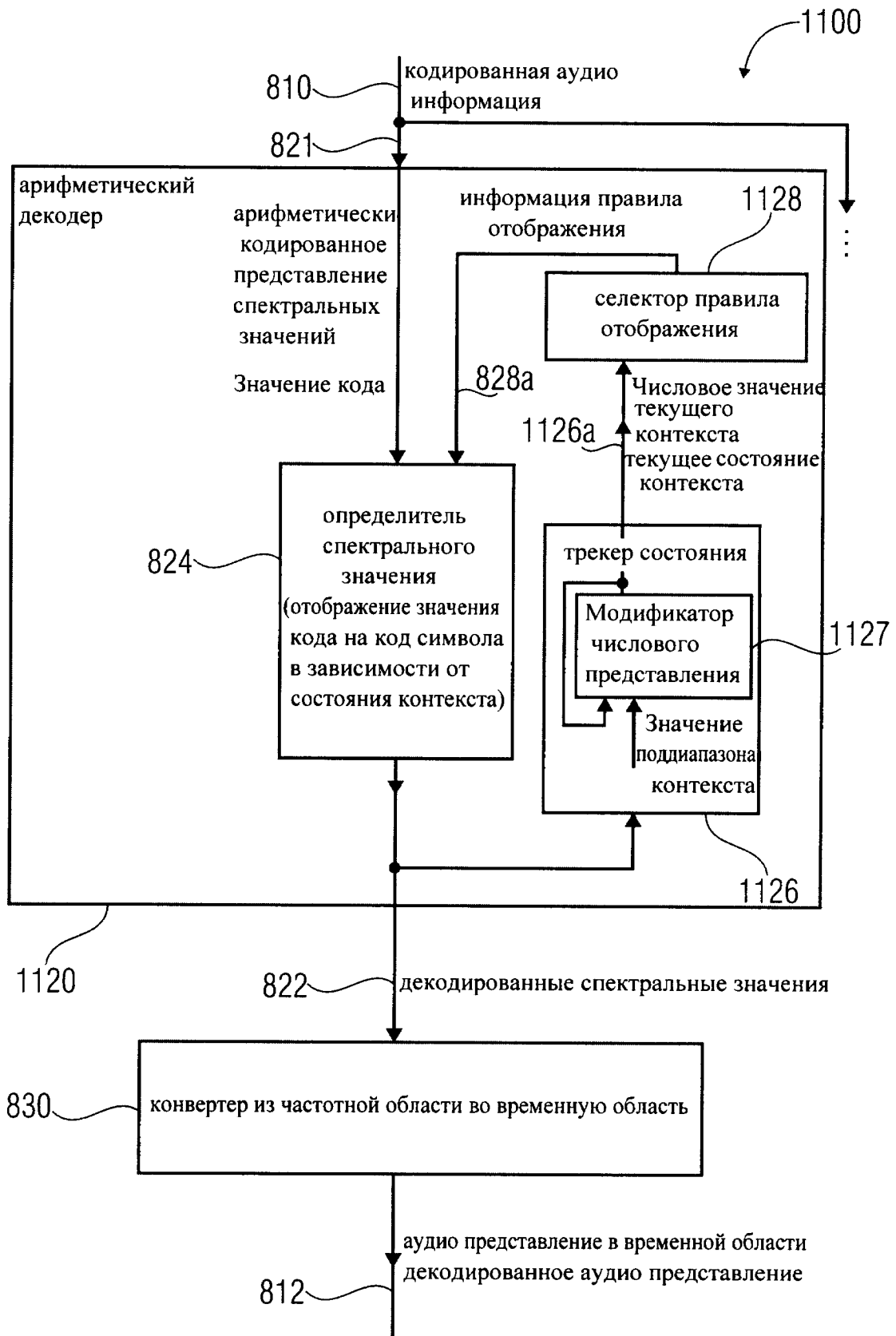
FIELD: physics.

SUBSTANCE: audio decoder includes: an arithmetic decoder for providing a plurality of decoded spectral values based on an arithmetic coded representation of the spectral values contained in the coded audio information and a frequency domain converter to the time domain for providing a time domain audio representation using the decoded spectral values in order to obtain decoded audio information; wherein the arithmetic decoder is configured to select a mapping rule describing the display of the code value

of the arithmetic coded representation of the spectral values to the symbol code; to determine the numerical value of the current context; and to modify the numerical representation of the previous context numerical value, depending on the value of the context subband, in order to obtain a numerical representation of the current context numerical value.

EFFECT: providing an improved compromise between the bit rate efficiency and the resource use efficiency.

20 cl, 67 dwg



ФИГ. 11

### Область применения

Воплощения в соответствии с изобретением связаны с аудио декодером для обеспечения декодированной аудио информации на основе кодированной аудио информации, аудио кодером для обеспечения кодированной аудио информации на основе входной аудио информации, способом для получения декодированной аудио информации на основе кодированной аудио информации, способом получения кодированной аудио информации на основе входной аудио информации и компьютерной программой.

Воплощения в соответствии с изобретением связаны с улучшенным спектральным бесшумным кодированием, которое может быть использовано в аудио кодере и декодере, например, в так называемом едином кодере речи и аудио (USAC).

### Уровень техники

Далее будет кратко описан уровень техники для того, чтобы облегчить понимание настоящего изобретения и его преимуществ. За последние десять лет были предприняты большие усилия для создания возможности цифрового хранения и распространения аудио содержания с хорошей эффективностью битрейта. Одним из важных достижений на этом пути является определение международного стандарта ISO/IEC 14496-3. Часть 3 данного стандарта связана с кодированием и декодированием аудио содержания, а подраздел 4 части 3 связан с общим аудио кодированием. ISO/IEC 14496, часть 3, раздел 4 определяет концепцию кодирования и декодирования общего аудио содержания. Кроме этого были предложены дальнейшие улучшения с целью повышения качества и/или снижения необходимой скорости передачи данных.

Согласно концепции, описанной в указанном стандарте, звуковой сигнал во временной области преобразуется в частотно-временное представление. Преобразование из временной области в частотно-временную область, как правило, осуществляется с помощью блоков преобразования, которые обозначаются как "кадры" образцов временной области. Было установлено, что выгоднее использовать перекрывающиеся кадры, которые перемещаются, например, на половину кадра, так как перекрытие позволяет эффективно избежать (или хотя бы уменьшить) артефакты. Кроме того, было обнаружено, что необходимо выполнить разделение на окна для того, чтобы избежать артефактов, появляющихся из-за обработки временно ограниченных кадров.

При преобразовании разделенной на окна части входного аудио сигнала из временной области в частотно-временную область, во многих случаях происходит уплотнение энергии, так что некоторые спектральные значения составляют значительно большую величину, чем множество других спектральных значений. Соответственно, во многих случаях есть сравнительно небольшое число спектральных значений с величиной, которая существенно выше средней величины спектральных значений. Типичным примером преобразования из временной области в частотно-временную область, приводящего к уплотнению энергии, является так называемое модифицированное дискретное косинус преобразование (MDCT).

Спектральные значения часто масштабируются и квантуются в соответствии с психоакустической моделью так, что ошибки квантования сравнительно меньше для психоакустически важных спектральных значений и сравнительно больше для психоакустически менее важных спектральных значений. Масштабированные и квантованные спектральные значения кодируются в целях обеспечения эффективного битрейта их представления.

Например, использование так называемого кодирования Хаффмана квантованных спектральных коэффициентов описано в международном стандарте ISO/IEC 14496-3:

2005 (E), часть 3, раздел 4.

Тем не менее, было установлено, что качество кодирования спектральных значений оказывает значительное влияние на требуемый битрейт. Кроме того, было установлено, что сложность аудио декодера, который часто представляет собой портативное устройство, используемое потребителем, и который поэтому должен быть дешевым и потреблять мало энергии, зависит от способа кодирования, используемого для кодирования спектральных значений.

В связи с этим, существует необходимость в разработке концепции кодирования и декодирования аудио содержания, которая предусматривает улучшение компромисса между эффективностью битрейта и эффективностью использования ресурсов.

Сущность изобретения

Примером воплощения изобретения является аудио декодер для получения декодированной аудио информации на основе кодированной аудио информации. Аудио декодер включает в себя арифметический декодер для получения множества декодированных спектральных значений на основе арифметически-кодированного представления спектральных значений. Аудио декодер также включает конвертер частотной области во временную область для получения аудио представления во временной области с помощью декодированных спектральных значений в целях получения декодированной аудио информации. Арифметический декодер предназначен для выбора правила отображения, описывающего отображение значения символа на код символа (код символа обычно описывает спектральное значение или множество спектральных значений или наиболее значимую битовую плоскость спектрального значения или множества спектральных значений) в зависимости от состояния контекста, которое описано с помощью числового значения текущего контекста. Арифметический декодер настроен определять числовое значение текущего контекста в зависимости от множества ранее декодированных спектральных значений.

Арифметический декодер настроен модифицировать числовое представление числового значения предыдущего контекста, которое описывает контекст, соответствующий одному или более ранее декодированному спектральному значению (а именно, описывает состояние контекста для декодирования указанного одного или более ранее декодированного спектрального значения), в зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, соответствующее одному или более декодируемому спектральному значению (или, более точно, описывает состояние контекста для декодирования указанного одного или более декодируемого спектрального значения).

Данный вариант реализации изобретения основан на том, что с точки зрения вычислений представляется эффективным модифицировать числовое представление числового значения текущего контекста в зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения текущего контекста, так как можно избежать полного повторного вычисления числового значения текущего контекста. Наоборот, можно применять корреляции между числовым значением предыдущего контекста и числовым значением текущего контекста для того, чтобы затраты на вычисления оставались сравнительно небольшими. Было обнаружено, что существует большой выбор возможностей для модификации числового представления числового значения текущего контекста, включая комбинацию повторного масштабирования числового представления числового значения текущего контекста, добавление значения поддиапазона контекста или извлеченного из него

значения (как, например, вариант значения поддиапазона контекста с битовым сдвигом) к числовому представлению числового значения текущего контекста или обработанному числовому представлению числового значения предыдущего контекста, замена части числового представления (а не числового представления полностью) числового значения  
5 предыдущего контекста в зависимости от значения поддиапазона контекста и т.д. Таким образом, сохранение по меньшей мере части числового представления числового значения предыдущего контекста (возможно вариант со сдвигом) позволяет значительно снизить вычислительные затраты на обновление числового значения контекста.

В предпочтительном варианте реализации изобретения арифметический декодер  
10 настроен обеспечивать числовое представление числового значения текущего контекста таким образом, что части числового представления, имеющие различный числовой вес, определяются различными значениями поддиапазона контекста. Соответственно, повторное обновление числового значения контекста для того, чтобы извлечь числовое значение текущего контекста из числового значения предыдущего контекста, может  
15 выполняться с небольшими вычислительными затратами из-за опущения большого количества информации.

В предпочтительном варианте реализации изобретения числовое представление является двоичным числовым представлением одного числового значения текущего контекста. Предпочтительно, первое подмножество битов двоичного числового  
20 представления определяется первым значением поддиапазона контекста, соответствующим одному или более ранее декодированному спектральному значению, и второе подмножество битов двоичного числового представления определяется вторым значением поддиапазона контекста, которое соответствует одному или более ранее декодированному спектральному значению, при этом биты первого подмножества  
25 битов имеют отличный от битов второго подмножества битов числовой вес. Было обнаружено, что такое представление подходит для повторного извлечения числового значения текущего контекста из числового значения предыдущего контекста.

В предпочтительном варианте реализации изобретения арифметический декодер настроен модифицировать поразрядное маскированное подмножество информационных  
30 битов числового представления числового значения предыдущего контекста либо вариант с битовым сдвигом числового представления числового значения предыдущего контекста в зависимости от значения поддиапазона контекста, который был учтен для извлечения числового значения предыдущего контекста, для того, чтобы получить числовое представление числового значения текущего контекста. При выполнении  
35 поразрядной маскировки числового представления числового значения текущего контекста или при выполнении битового сдвига числового представления числового значения текущего контекста можно достичь того, что части контекста, которые уже не так релевантны, как ранее, удаляются из числового значения контекста и, предпочтительно, заменяются другими частями контекста, более релевантными в  
40 текущем контексте. Поразрядная маскировка подмножества информационных битов числового представления числового значения предыдущего контекста позволяет заменить часть числового значения предыдущего контекста в зависимости от значения поддиапазона контекста, которое, в свою очередь, позволяет учесть часть контекста, которая ранее еще не учитывалась. Кроме этого, операция сдвига отражает факт  
45 наложения ранее декодированных спектральных значений, используемых для определения предыдущего контекста (т.е. контекста, который используется для декодирования предыдущего кортежа спектральных значений), и ранее декодированных спектральных значений, используемых для определения текущего контекста (т.е.

контекста, который используется для декодирования спектральных значений, которые декодируются в настоящий момент). Кроме этого, операции сдвига отражают тот факт, что отношения по частоте (например, равны по частоте, превышают частоту на один частотный поддиапазон и т.д.) ранее декодированных спектральных значений по отношению к спектральным значениям, декодированным с использованием числового значения предыдущего контекста, отличаются от частотных отношений ранее декодированных спектральных значений по отношению к спектральным значениям, которые должны быть декодированы с использованием числового значения текущего контекста.

10 В предпочтительном варианте реализации изобретения арифметический декодер настроен на выполнение битового сдвига числового представления числового значения предыдущего контекста таким образом, что числовой вес битовых подмножеств, соответствующих различным значениям поддиапазона контекста, подвергается модификации, чтобы получить числовое представление числового значения текущего контекста. Соответственно, сдвиг частотной позиции между одним или более спектральным значением, декодированным при помощи числового значения предыдущего контекста и одним или более спектральным значением, которое должно быть декодировано при помощи числового значения текущего контекста может эффективным образом отражаться в числовом значении контекста. Кроме этого, операция сдвига обычно может выполняться при низких затратах на вычисления с использованием стандартного микропроцессора.

В предпочтительном варианте реализации изобретения арифметический декодер настроен на выполнение битового сдвига числового представления числового значения предыдущего контекста таким образом, что битовое подмножество, соответствующее значению поддиапазона контекста, удаляется из числового представления, чтобы получить числовое представление числового значения текущего контекста. Соответственно, единичная операция сдвига выполняет двойную функцию, а именно учитываются изменения частотного положения, а также учитывается тот факт, что некоторые спектральные значения (представленные с помощью значения поддиапазона контекста), которые использовались для получения числового значения предыдущего контекста, далее не нужны для получения числового значения текущего контекста.

В предпочтительном варианте реализации изобретения арифметический декодер настроен модифицировать первое битовое подмножество двоичного числового представления числового значения предыдущего контекста или вариант с битовым сдвигом двоичного числового представления числового значения предыдущего контекста в зависимости от значения поддиапазона контекста и оставлять без изменений вторые битовые подмножества двоичного числового представления числового значения предыдущего контекста или вариант с битовым сдвигом двоичного числового представления числового значения предыдущего контекста, а также извлекать двоичное числовое представление числового значения текущего контекста из двоичного числового представления числового значения предыдущего контекста при помощи выборочного модифицирования одного или более битового подмножества, соответствующего поддиапазонам контекста, которые учитываются для декодирования ранее декодированных спектральных значений (декодированных при помощи числового значения предыдущего контекста) и не учитываются для декодирования текущих спектральных значений при помощи числового значения текущего контекста. Была доказана особая эффективность этого подхода.

В предпочтительном варианте реализации изобретения арифметический декодер



настроен обеспечивать числовое представление числового значения текущего контекста таким образом, что подмножество наименее значимых битов числового представления числового значения текущего контекста описывает значение поддиапазона контекста, которое используется для декодирования спектральных значений, для которых  
5 определяется состояние контекста при помощи числового значения текущего контекста, и значение поддиапазона контекста которого не используется для декодирования спектральных значений, для которых определяется состояние контекста при помощи  
10 числового значения последующего контекста (напр., числовое значение контекста, извлеченное из числового значения текущего контекста). Такой подход позволяет извлечь числовое значение текущего контекста из числового значения предыдущего  
15 контекста (а также извлечь числовое значение последующего контекста из числового значения текущего контекста) при помощи операции сдвига, т.к. наименее значимые биты числового представления могут легко переходить в верхний регистр. Кроме этого, было обнаружено, что целесообразно назначать небольшой числовой вес значениям  
20 поддиапазона контекста, которые релевантны для числового значения предыдущего контекста, но уже не релевантны для числового значения текущего контекста (или, аналогично, релевантны для числового значения текущего контекста, но уже не релевантны для числового значения последующего контекста), так как это позволяет выполнить эффективное отображение числового значения (текущего) контекста на  
значение индекса правила отображения.

В предпочтительном варианте реализации изобретения арифметический декодер настроен оценивать по меньшей мере одну таблицу, чтобы определить, идентично ли  
числовое значение текущего контекста табличному значению контекста (например, значимому значению состояния), которое описано с помощью записи таблицы, или  
25 находится в пределах интервала, описанного с помощью записей таблицы, а также извлекать значение индекса правила отображения, которое описывает выбранное правило отображения в зависимости от результата оценки по меньшей мере одной  
30 таблицы. Было обнаружено, что числовое значение (текущего) контекста, которое формируется и обновляется как было рассмотрено выше, хорошо подходит для такого отображения на значение индекса правила отображения.

В предпочтительном варианте реализации изобретения арифметический декодер настроен контролировать, является ли сумма множества значений поддиапазона  
контекста меньшей или равной заданному пороговому значению суммы, а также выборочно модифицировать числовое значение текущего контекста в зависимости от  
35 результата проверки. Было обнаружено, что такая дополнительная выборочная модификация числового значения текущего контекста хорошо подходит для эффективного ввода значимой информации о контексте в числовое значение текущего  
контекста без какого-либо противоречия по отношению к концепции обновления  
числового значения контекста.

В предпочтительном варианте реализации изобретения арифметический декодер настроен контролировать, является ли сумма множества значений поддиапазона  
контекста, значения поддиапазона контекста которой соответствуют той же временной  
40 части аудио содержания, что и одно или более спектральное значение, декодируемое с использованием состояния контекста, определяемого с помощью числового значения  
текущего контекста, а также значения поддиапазона контекста которой соответствуют  
45 более низким частотам, чем одно или более спектральное значение, декодируемое с использованием состояния контекста, определяемого с помощью числового значения  
текущего контекста, меньшей или равной заданному пороговому значению суммы, а

также выборочно модифицировать числовое значение текущего контекста в зависимости от результата проверки. Было обнаружено, что такая проверка присутствия области сравнительно небольших спектральных значений обеспечивает ценную дополнительную информацию.

5 В предпочтительном варианте реализации изобретения арифметический декодер настроен суммировать абсолютные значения первого множества ранее декодированных спектральных значений, чтобы получить первое значение поддиапазона контекста, соответствующее первому множеству ранее декодированных спектральных значений, а также суммировать абсолютные значения второго множества ранее декодированных спектральных значений, чтобы получить второе значение поддиапазона контекста, соответствующее второму множеству ранее декодированных спектральных значений. Соответственно, могут быть получены различные значения поддиапазона контекста.

10 В предпочтительном варианте реализации изобретения арифметический декодер настроен ограничивать значения поддиапазона контекста таким образом, что значения поддиапазона контекста оказываются репрезентативными при использовании действительного подмножества информационных битов числового представления числового значения предыдущего контекста. Было обнаружено, что ограничение значений поддиапазона контекста не оказывает отрицательное влияние на информационное содержание значений поддиапазона контекста. Однако, такое

15 ограничение имеет преимущество в том, что количество битов, необходимых для представления значения поддиапазона контекста может оставаться достаточно небольшим что оказывает положительное влияние с точки зрения необходимого объема памяти. Также ограничение значений поддиапазона контекста способствует повторному обновлению числового значения контекста.

25 Следующий вариант реализации изобретения создает аудио кодер для обеспечения кодированной информации на основе входной аудио информации. Аудио кодер содержит энергосберегающий конвертер временной области в частотную область для обеспечения аудио представления в частотной области на основе представления входной аудио информации во временной области таким образом, что аудио представление частотной

30 области включает набор спектральных значений. Аудио кодер также включает арифметический кодер, которые настроен кодировать спектральное значение или его обработанную ранее версию, либо - аналогично-множество спектральных значений или их обработанных ранее версий, используя кодовое слово переменной длины.

35 Арифметический кодер настроен отображать спектральное значение или значение наиболее значимой битовой плоскости спектрального значения на значение кода. Арифметический кодер настроен выбирать правило отображения, которое описывает отображение спектрального значения или значения наиболее значимой битовой плоскости спектрального значения на значение кода в зависимости от состояния контекста, описанного с помощью числового значения текущего контекста.

40 Арифметический кодер настроен определять числовое значение текущего контекста в зависимости от множества ранее кодированных спектральных значений.

Арифметический кодер настроен модифицировать числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее одному или более ранее кодированному спектральному значению

45 (или, более точно, описывает контекст для кодирования указанного одного или более ранее кодированного спектрального значения), в зависимости от значения поддиапазона контекста, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, соответствующее одному или более

кодируемому спектральному значению (или, более точно, описывает состояние контекста для кодирования указанного одного или более кодируемого спектрального значения).

Аудио кодер основан на тех же идеях, что и аудио декодер. Также аудио кодер быть дополнен теми же функциями, которые рассматривались по отношению к аудио декодеру.

Следующий вариант реализации изобретения создает способ для обеспечения декодированной аудио информации на основе кодированной аудио информации.

Следующий вариант реализации изобретения создает способ для обеспечения кодированной аудио информации на основе входной аудио информации.

Следующий вариант реализации изобретения создает компьютерную программу для выполнения одного из указанных способов.

Краткое описание чертежей

Варианты реализации настоящего изобретения будут далее описаны со ссылкой на прилагаемые фигуры:

Фиг. 1 показывает блок-схему аудио кодера согласно одному из вариантов реализации изобретения;

Фиг. 2 показывает блок-схему аудио декодера согласно одному из вариантов реализации изобретения;

Фиг. 3 показывает представление кода псевдо-программы алгоритма "values\_decode ()» для декодирования спектральных значений;

Фиг. 4 показывает схематическое представление контекста для вычисления состояния;

Фиг. 5a показывает представление кода псевдо-программы алгоритма "arith\_map\_context()» для отображения контекста;

Фиг. 5b показывает представление кода псевдо-программы следующего алгоритма "arith\_map\_context()» для отображения контекста;

Фиг. 5c показывает представление кода псевдо-программы алгоритма "arith\_get\_context()» для получения значения состояния контекста;

Фиг. 5d показывает представление кода псевдо-программы алгоритма "arith\_get\_context()» для получения значения состояния контекста;

Фиг. 5e показывает представление кода псевдо-программы алгоритма "arith\_get\_pk ()» для извлечения значения индекса сводной таблицы частот „pk<sub>i</sub>” из значения состояния (или переменной состояния);

Фиг. 5f показывает представление кода псевдо-программы алгоритма "arith\_get\_pk ()» для извлечения значения индекса сводной таблицы частот „pk<sub>i</sub>” из значения состояния (или переменной состояния);

Фиг. 5g показывает представление кода псевдо-программы алгоритма "arith\_decode ()» для арифметического декодирования символа из кодового слова переменной длины;

Фиг. 5h показывает первую часть представления кода псевдо-программы следующего алгоритма "arith\_decode()» для арифметического декодирования символа из кодового слова переменной длины;

Фиг. 5i показывает вторую часть представления кода псевдо-программы следующего алгоритма "arith\_decode()» для арифметического декодирования символа из кодового слова переменной длины;

Фиг. 5j показывает представление кода псевдо-программы алгоритма для извлечения абсолютных величин a, b спектральных значений из общего значения m;

Фиг. 5k показывает представление кода псевдо-программы алгоритма для ввода декодированных значений a, b в массив декодированных спектральных значений;

Фиг. 5l показывает представление кода псевдо-программы алгоритма

"arith\_update\_context()» для получения значения контекста поддиапазона основе абсолютных значений a, b декодированных спектральных значений;

Фиг. 5m показывает представление кода псевдо-программы алгоритма "arith\_finish ()» для заполнения записей массива декодированных спектральных значений и массива значений контекста поддиапазона;

Фиг. 5n показывает представление кода псевдо-программы следующего алгоритма для извлечения абсолютных значений a, b декодированных спектральных значений из общего значения m;

Фиг. 5o показывает представление кода псевдо-программы алгоритма "arith\_update\_context()" для обновления массива декодированных спектральных значений и массива значений контекста поддиапазона;

Фиг. 5p показывает представление кода псевдо-программы алгоритма "arith\_save\_context()» для заполнения записей массива декодированных спектральных значений и массива значений контекста поддиапазона;

Фиг. 5q показывает легенду определений;

Фиг. 5r показывает легенду переменных;

Фиг. 6a показывает синтаксис представления необработанного блока данных единого кодирования речи и аудио (USAC);

Фиг. 6b показывает синтаксис представления единого элемента канала;

Фиг. 6c показывает синтаксис представления парного элемента канала;

Фиг. 6d показывает синтаксис представления контрольной информации "ICS";

Фиг. 6e показывает синтаксис представления потока канала частотной области;

Фиг. 6f показывает синтаксис представления арифметически кодированных спектральных данных;

Фиг. 6g показывает синтаксис представления для декодирования набора спектральных значений;

Фиг. 6h показывает следующий синтаксис представления для декодирования набора спектральных значений;

Фиг. 6i показывает легенду элементов данных и переменных;

Фиг. 6j показывает следующую легенду элементов данных и переменных;

Фиг. 7 показывает блок-схему аудио кодера согласно первому аспекту изобретения;

Фиг. 8 показывает блок-схему аудио декодера согласно первому аспекту изобретения;

Фиг. 9 показывает графическое представление отображения числового значения текущего контекста на значение индекса правила отображения согласно первому аспекту изобретения;

Фиг. 10 показывает блок-схему аудио кодера согласно второму аспекту изобретения;

Фиг. 11 показывает блок-схему аудио декодера согласно второму аспекту изобретения;

Фиг. 12 показывает блок-схему аудио кодера согласно третьему аспекту изобретения;

Фиг. 13 показывает блок-схему аудио декодера согласно третьему аспекту изобретения;

Фиг. 14a показывает схематическое представление контекста для вычисления состояния как оно используется в соответствии с рабочим проектом 4 проекта стандарта USAC;

Фиг. 14b показывает обзор таблиц, используемых в схеме арифметического кодирования в соответствии с рабочим проектом 4 проекта стандарта USAC;

Фиг. 15a показывает схематическое представление контекста для вычисления состояния как оно используется в вариантах реализации настоящего изобретения

Фиг. 15b показывает обзор таблиц, используемых в схеме арифметического кодирования согласно настоящему изобретению;

Фиг. 16a показывает графическое представление запроса памяти только для чтения на схемы бесшумного кодирования в соответствии с настоящим изобретением, рабочим проектом 5 проекта стандарта USAC и согласно кодированию Хаффмана AAC (усовершенствованное аудио кодирование);

Фиг. 16b показывает графическое представление общего запроса данных памяти только для чтения декодера USAC в соответствии с настоящим изобретением и в соответствии с рабочим проектом 5 проекта стандарта USAC;

Фиг. 17 показывает графическое представление порядка сравнения бесшумного кодирования в соответствии с рабочим проектом 3 или рабочим проектом 5 проекта стандарта USAC со схемой кодирования согласно настоящему изобретению;

Фиг. 18 показывает таблицу представления средних битрейтов арифметического кодера USAC в соответствии с рабочим проектом 3 проекта стандарта USAC и в соответствии с вариантом реализации настоящего изобретения;

Фиг. 19 показывает таблицу представления минимального и максимального уровней резервуара бит для арифметического декодера в соответствии с рабочим проектом 3 проекта стандарта USAC и для арифметического декодера в соответствии с вариантом реализации настоящего изобретения;

Фиг. 20 показывает таблицу представления обычных порядковых чисел для декодирования 32-кбитного потока в соответствии с рабочим проектом 3 проекта стандарта USAC для различных версий арифметического кодера;

Фиг. 21(1) и 21(2) показывают содержание таблицы "ari\_lookup\_m[600]";

Фиг. 22(1) - 21(4) показывают содержание таблицы "ari\_hash\_m[600]";

Фиг. 23(1) - 23(7) показывают содержание таблицы "ari\_cf\_m[600]";

Фиг. 24 показывает содержание таблицы "ari\_cf\_r[]".

Подробное описание вариантов реализации изобретения

1. Аудио кодер в соответствии с фиг. 7

Фиг. 7 показывает блок-схему аудио кодера, согласно одному из вариантов реализации изобретения. Аудио декодер 700 настроен получать входную аудио информацию 710 и получать на ее основе кодированную аудио информацию 712. Аудио кодер включает в себя энергосберегающий конвертер временной области в частотную область 720, который предназначен обеспечивать аудио представление в частотной области 722 на основе представления входной аудио информации 710 во временной области так, что аудио представление в частотной области 722 включает в себя набор спектральных значений. Аудио кодер 700 также включает в себя арифметический кодер 730, предназначенный для кодирования спектрального значения (из набора спектральных значений, формирующих аудио представление в частотной области 722) или его предварительно обработанной версии с помощью кодового слова переменной длины, чтобы получить кодированную аудио информацию 712 (которая может включать, например, множество кодовых слов переменной длины).

Арифметический кодер 730 настроен на отображение спектрального значения или значения наиболее значимой битовой плоскости спектрального значения на значение кода (т.е. на кодовое слово переменной длины) в зависимости от состояния контекста. Арифметический кодер предназначен для выбора правила отображения, описывающего отображение спектрального значения или наиболее значимой битовой плоскости спектрального значения на значение кода в зависимости от состояния (текущего) контекста. Арифметический кодер настроен, чтобы определять текущее состояние

контекста или числовое значение текущего контекста в зависимости от множества ранее кодированных (предпочтительно, но не обязательно смежных) спектральных значений. Для этого арифметический кодер настроен оценивать хэш-таблицу, записи которой определяют как значения значимого состояния среди числовых значений контекста, так и границы интервалов числовых значений контекста, при этом значение индекса правила отображения индивидуально соотносится с числовым значением (текущего) контекста, так как является значением значимого состояния, при этом общее значение индекса правила отображения соотносится с разными числовыми значениями (текущего) контекста, которые находятся в пределах интервала, ограниченного границами (при этом границы интервала предпочтительно определяются записями хэш-таблицы).

Как можно видеть, отображение спектрального значения (аудио представления в частотной плоскости 722) или наиболее значимой битовой плоскости спектрального значения на значение кода (кодированной аудио информации 712) может осуществляться кодированием спектрального значения 740 с помощью правила отображения 742.

Трекер состояния 750 может быть настроен для отслеживания состояния контекста. Трекер состояния 750 обеспечивает информацию 754, описывающую текущее состояние контекста. Информация 754, описывающая текущее состояние контекста, предпочтительно может быть в форме числового значения текущего контекста. Селектор правила отображения 760 настроен выбирать правило отображения, например, сводную таблицу частот, описывающую отображение спектрального значения, или наиболее значимую битовую плоскость спектрального значения, на значение кода.

Соответственно, селектор правила отображения 760 предоставляет информацию правила отображения 742 для кодирования спектрального значения 740. Информация правила отображения 742 может иметь форму значения индекса правила отображения или сводной таблицы частот, которая выбирается в зависимости от значения индекса правила отображения. Селектор правила отображения 760 включает (или по меньшей мере оценивает) хэш-таблицу 752, записи которой определяют как значения значимого состояния среди числовых значений контекста, так и границы интервалов числовых значений контекста, при этом значение индекса правила отображения индивидуально соотносится с числовым значением контекста, так как является значением значимого состояния, при этом общее значение индекса правила отображения соотносится с разными числовыми значениями (текущего) контекста, которые находятся в пределах интервала, ограниченного границами. Хэш-таблица 762 оценивается для того, чтобы выбрать правило отображения, т.е. чтобы обеспечить информацию правила отображения 742.

Подводя итог вышесказанному, аудио кодер 700 выполняет арифметическое кодирование аудио представления в частотной области, осуществляемого конвертером временной области в частотную область. Арифметическое кодирование зависит от контекста, например, правило отображения (например, сводная таблица частот) выбирается в зависимости от ранее кодированных спектральных значений. Таким образом, спектральные значения, смежные во времени и/или по частоте (или по меньшей мере в заданном окружении) друг с другом и/или с кодируемым в данный момент спектральным значением (т.е. спектральные значения в пределах заданного окружения в данный момент кодируемого спектрального значения) рассматриваются в арифметическом кодировании для регулировки распределения вероятности, оцениваемой арифметическим кодированием. При выборе соответствующего правила отображения оцениваются числовые значения текущего контекста 754, предоставленные трекером состояния 750. Поскольку обычно число различных правил отображения значительно

меньше, чем число возможных числовых значений текущего контекста 754, селектор правила отображения 760 определяет одни и те же правила отображения (описанные, например, с помощью значения индекса правила отображения) для сравнительно большого количества различных числовых значений контекста. Тем не менее, обычно существуют особые спектральные конфигурации (представленные особыми числовыми значениями контекста), которым должно соответствовать определенное правило отображения для того, чтобы обеспечить эффективное кодирование.

Обнаружено, что выбор правила отображения в зависимости от числового значения текущего контекста может осуществляться с особо высокой эффективностью вычисления, если записи одной хэш-таблицы определяют как значения значимого состояния, так и границы интервалов числовых значений (текущего) контекста. Было обнаружено, что этот механизм хорошо адаптируется к требованиям выбора правила отображения, так как во многих случаях единое значение значимого состояния (или значимое числовое значение контекста) находится между интервалом множества значений незначимого состояния, который расположен слева (которому соответствует общее правило отображения) и интервалом множества значений незначимого состояния, который расположен справа (которому соответствует общее правило отображения). Также механизм использования единой хэш-таблицы, записи которой определяют как значения значимого состояния, так и границы интервалов числовых значений (текущего) контекста может эффективно справляться с различными случаями, когда, например, существуют два смежных интервала значений незначимого состояния (также обозначенных как незначимые числовые значения состояния) без значения значимого состояния между ними. Высокая эффективность вычисления достигается благодаря малому количеству обращений к таблице. Например, единичный итеративный табличный поиск оказывается достаточным в большинстве вариантов реализации изобретения для того, чтобы обнаружить, равняется ли числовое значение текущего контекста одному из значений значимого состояния, либо в каком из интервалов значений незначимого состояния находится числовое значение текущего контекста.

Соответственно, число обращений к таблице, которые являются энергоемкими и затратными по времени, останется небольшим. Таким образом, селектор правила отображения 760, который использует хэш-таблицу 762, может рассматриваться как эффективный селектор правила отображения с точки зрения сложности вычисления, но при этом позволяющий добиться эффективного кодирования (в плане битрейта).

Дальнейшие детали относительно извлечения информации о правиле отображений 742 из числового значения текущего контекста 754 будут рассмотрены ниже.

## 2. Аудио декодер в соответствии с фиг. 8

Фиг. 8 показывает блок-схему аудио декодера 800. Аудио декодер 800 настроен на получение закодированной аудио информации 810 и на представлении на ее основе декодированной аудио информации 812. Аудио декодер 800 включает в себя арифметический декодер 820, который предназначен для предоставления множества декодированных спектральных значений 822 на основе арифметически закодированного представления 821 спектральных значений. Аудио декодер 800 также включает конвертер частотной области во временную область 830, который предназначен для получения декодированных спектральных значений 822 и получения аудио представления во временной области 812, которое может включать декодированную аудио информацию, с помощью декодированных спектральных значений 822, для получения декодированной аудио информации 812.

Арифметический декодер 820 включает в себя определитель спектрального значения

824, настроенный на отображение значения кода арифметически кодированного представления 821 спектральных значений на код символа, представляющий одно или более декодированных спектральных значений, или по меньшей мере часть (например, наиболее значимую битовую плоскость) одного или более декодированных спектральных значений. Определитель спектрального значения 824 может быть настроен для выполнения отображения в зависимости от правила отображения, которое может быть описано в информации правила отображения 828а. Информация правила отображения 828а может быть, например, в форме значения индекса правила отображения или выбранной сводной таблицы частот (которая выбирается, например, в зависимости от значения индекса правила отображения).

Арифметический декодер 820 настроен на выбор правила отображения (например, сводной таблицы частот), описывающего отображение значений кода (которое описывается с помощью арифметически кодированного представления 821 спектральных значений) на код символа (описывающий одно или более спектральных значений или его наиболее значимую битовую плоскость) в зависимости от состояния контекста (которое может быть описано в информации состояния контекста 826а). Арифметический декодер 820 настроен определять текущее состояние контекста (описанное с помощью числового значения текущего контекста) в зависимости от множества ранее декодированных спектральных значений. Для этого может быть использован трекер состояния 826, который получает информацию с описанием ранее декодированных спектральных значений и обеспечивает на их основе числовое значение текущего контекста 826а, которое описывает состояние текущего контекста.

Арифметический декодер также настроен оценивать хэш-таблицу 829, записи которой определяют как значения значимого состояния среди числовых значений контекста, так и границы интервалов числовых значений контекста для того, чтобы выбрать правило отображения, при этом значение индекса правила отображения индивидуально соотносится с числовым значением контекста, так как является значением значимого состояния, при этом общее значение индекса правила отображения соотносится с разными числовыми значениями контекста, которые находятся в пределах интервала, ограниченного границами. Оценка хэш-таблицы 829 может осуществляться, например, при помощи блока оценки хэш-таблицы, который может быть частью селектора правила отображения 828. Соответственно, информация правила отображения 828а, например, в форме значения индекса правила отображения может быть получена на основе числового значения текущего контекста 826а, которое описывает состояние текущего контекста. Селектор правила отображения 828 может, например, определять значение индекса правила отображения 828а в зависимости от результата оценки хэш-таблицы 829. В качестве альтернативы оценка хэш-таблицы 829 может напрямую обеспечивать получение значения индекса правила отображения.

Что касается функциональных возможностей декодера аудио сигнала 800, следует отметить, что арифметический декодер 820 настроен на выбор правила отображения (например, сводную таблицу частот), которое, в среднем, хорошо адаптировано к декодируемым спектральным значениям, так как правило отображения выбирается в зависимости от состояния текущего контекста (описанного, например, с помощью числового значения текущего состояния), что, в свою очередь, определяется в зависимости от множества ранее декодированных спектральных значений. Таким образом, могут быть использованы статистические зависимости между декодируемыми смежными спектральными значениями. Кроме того, при использовании селектора правила отображения 828 может быть эффективно реализован арифметический декодер



820, обеспечивающий компромисс между сложностью вычислений, размером таблицы и эффективностью кодирования. При оценке (единичной) хэш-таблицы 829, записи которой описывают как значения значимого состояния, так и границы интервалов значений незначимого состояния, может быть достаточно единичного итеративного табличного поиска для того, чтобы извлечь информацию правила отображения 828а из числового значения текущего контекста 826а. Соответственно, становится возможным отображать сравнительно большое число различных возможных числовых значений (текущего) контекста на сравнительно небольшое количество различных значений индекса правила отображения. Используя хэш-таблицу 829, как было описано выше, возможно применить то, что во многих случаях единичное изолированное значение значимого состояния (значение значимого контекста) находится между интервалом значений незначимого состояния, который расположен слева (значения незначимого контекста) и интервалом значений незначимого состояния, который расположен справа (значения незначимого контекста), при этом другое значение индекса правила отображения соотносится со значением значимого состояния (значение значимого контекста), при сравнении значений состояния (значений контекста) левостороннего интервала и значений состояния (значений контекста) правостороннего интервала состояния. Однако использование хэш-таблицы 829 также хорошо подходит для ситуаций, когда два интервала числовых значений непосредственно примыкают друг к другу без значения значимого состояния между ними.

Итак, селектор правила отображения 828, который оценивает хэш-таблицу 829, является особенно эффективным при выборе правила отображения (или в случае предоставления значения индекса правила отображения) в зависимости от состояния текущего контекста (или в зависимости от числового значения текущего контекста, описывающего состояние текущего контекста), так как механизм хеширования хорошо адаптируется к типичным сценариям контекста в аудио декодере.

Дальнейшие подробности будут описаны ниже.

### 3. Механизм хеширования значения контекста в соответствии с фиг. 8

Далее будет рассмотрен механизм хеширования контекста, который может быть реализован селектором правила отображения 760 и/или селектором правила отображения 828. Для того чтобы реализовать указанный механизм хеширования значения контекста может быть использована хэш-таблица 762 и/или хэш-таблица 829.

Далее будут рассмотрены детали сценария хеширования числового значения текущего контекста со ссылкой на фиг. 9. Как показывает график на фиг. 9, абсцисса 910 представляет числовые значения текущего контекста (т.е. числовые значения контекста). Ордината 912 представляет значения индекса правила отображения. Маркеры 914 представляют значения индекса правила отображения для незначимых числовых значений контекста (которые описывают незначимые состояния). Маркеры 916 описывают значения индекса правила отображения для «индивидуальных» (действительных) значимых числовых значений контекста, описывающих индивидуальные (действительные) значимые состояния. Маркеры 916 описывают значения индекса правила отображения для «неподходящих» числовых значений контекста, которые описывают «неподходящие» значимые состояния, при этом «неподходящим» значимым состоянием является значимое состояние, которому соответствует то же значение индекса правила отображения, что и одному из смежных интервалов незначимых числовых значений контекста.

Как можно увидеть, запись хэш-таблицы "ari\_hash\_m[i1]" описывает индивидуальное (действительное) значимое состояние, имеющее числовое значение контекста c1. Как

можно увидеть, значение индекса правила отображения `nriv1` соотносится с индивидуальным (действительным) значимым состоянием, которое имеет числовое значение контекста `c1`. Соответственно, как числовое значение контекста `c1`, так и индекса правила отображения `nriv1` могут быть описаны записью хэш-таблицы `"ari_hash_m[i1]"`. Интервал 932 числовых значений контекста ограничен числовым значением контекста `c1`, при этом числовое значение контекста `c1` не принадлежит интервалу 932, так что самое большое числовое значение контекста интервала 932 равно `c1-1`. значение индекса правила отображения `nriv4` (которое отличается от `nriv1`) соответствует числовым значениям контекста интервала 932. значение индекса правила отображения `nriv4` может быть описано, например, с помощью записи таблицы `"ari_lookup_m[i1-1]"` дополнительной таблицы `"ari_lookup_m"`.

Кроме этого, значение индекса правила отображения `nriv2` может соответствовать числовым значениям контекста, находящимся в пределах интервала 934. Нижняя граница интервала 934 определяется с помощью числового значения контекста `c1`, которое является значимым числовым значением контекста, при этом числовое значение контекста `c1` не принадлежит интервалу 932. Соответственно, наименьшее значение интервала 934 равно `c1+1` (исходя из предположения, что числовые значения контекста являются целыми числами). Другая граница интервала 934 определяется с помощью числового значения контекста `c2`, при этом числовое значение контекста не принадлежит интервалу 934, так что наибольшее значение интервала 934 равно `c2-1`. Числовое значение контекста `c2` является так называемым «неподходящим» числовым значением контекста, которое описывается с помощью записи хэш-таблицы `"ari_hash_m[i2]"`. Например, значение индекса правила отображения `nriv2` может соотноситься с числовым значением контекста `c2` таким образом, что числовое значение контекста, соответствующее «неподходящему» значимому числовому значению контекста `c2`, равно значению индекса правила отображения, соответствующему интервалу 934, ограниченному числовым значением контекста `c2`. Кроме этого, интервал 936 числового значения контекста также ограничен числовым значением контекста `c2`, при этом числовое значение контекста `c2` не принадлежит интервалу 936, так что наименьшее числовое значение контекста интервала 936 равно `c2+1`. Значение индекса правила отображения `nriv3`, которое обычно отличается от значения индекса правила отображения `nriv2`, соответствует числовым значениям контекста интервала 936.

Как можно увидеть, значение индекса правила отображения `nriv4`, которое соответствует интервалу 932 числовых значений контекста, может быть описано с помощью записи `"ari_lookup_m[i1-1]"` таблицы `"ari_lookup_m"`, индекс правила отображения `nriv2`, соответствующий числовым значениям контекста интервала 934, может быть описано с помощью записи `"ari_lookup_m[i1]"` таблицы `"ari_lookup_m"`, и значение индекса правила отображения `nriv3` может быть описано с помощью записи `"ari_lookup_m[i2]"` таблицы `"ari_lookup_m"`. В данном здесь примере значение индекса хэш-таблицы `i2`, может быть больше на 1, чем значение индекса хэш-таблицы `i1`.

Как можно увидеть на фиг. 9, селектор правила отображения 760 или селектор правила отображения 828 может получать числовое значение текущего контекста 764, 826а и решать при помощи оценки записей таблицы `"ari_hash_m"`, является ли числовое значение текущего контекста значимым значением состояния (независимо от того, является ли оно «индивидуальным» значимым значением состояния или «неподходящим» значимым значением состояния) или числовое значение текущего контекста находится в пределах одного из интервалов 932, 934, 936, которые ограничены («индивидуальным» или «неподходящим») значением значимого состояния `c1`, `c2`. Как проверка, равняется

ли числовое значение текущего контекста значению значимого состояния  $s_1$ ,  $s_2$ , так и оценка, в каком из интервалов 932, 934, 936 находится числовое значение текущего контекста (в том случае, если числовое значение текущего контекста не равно значению значимого состояния) может осуществляться при использовании табличного поиска

5 единичной, общей хэш-таблицы.

Кроме этого, можно использовать оценку хэш-таблицы "ari\_hash\_m" для получения значения индекса хэш-таблицы (например,  $i_1-1$ ,  $i_1$ ,  $i_2$ ). Таким образом, селектор правила отображения 760, 828 настроен получать, с помощью оценки единичной хэш-таблицы 762, 829 (например, хэш-таблицы "ari\_hash\_m"), значение индекса хэш-таблицы

10 (например,  $i_1-1$ ,  $i_1$  или  $i_2$ ), обозначающий значение значимого состояния (например,  $s_1$  или  $s_2$ ) и/или интервал (например, 932, 934, 936), а также информацию о том, является ли числовое значение текущего контекста значимым значением контекста (также обозначенным как значимое значение состояния) или нет.

Кроме этого, если в ходе оценки хэш-таблицы 762, 829 "ari\_hash\_m" обнаруживается,

15 что числовое значение текущего контекста не является «значимым» значением контекста (или «значимым» значением состояния), может использоваться значение индекса хэш-таблицы (например,  $i_1-1$ ,  $i_1$  или  $i_2$ ), полученное при оценке хэш таблицы "ari\_hash\_m" для получения значения индекса правила отображения, соответствующего интервалам 932, 934, 936 числовых значений контекста. Например, значение индекса хэш-таблицы

20 (например,  $i_1-1$ ,  $i_1$  или  $i_2$ ) может использоваться для обозначения записи дополнительной таблицы отображения (например, "ari\_lookup\_m"), которая описывает значения индекса правила отображения, соответствующие интервалам 932, 934, 936, в пределах которых находится числовое значение текущего контекста.

Дальнейшие детали будут рассмотрены далее при подробном обсуждении алгоритма

25 "arith\_get\_pk" (при этом существуют различные варианты алгоритма "arith\_get\_pk"), примеры которого показаны на фиг. 5e и 5f.

Кроме этого, необходимо отметить, что размер интервалов может отличаться в каждом отдельном случае. В некоторых случаях интервал числовых значений контекста включает единичное числовое значение контекста. Однако во многих случаях интервал

30 включает множество числовых значений контекста.

#### 4. Аудио кодер в соответствии с фиг. 10

На фиг. 10 показана блок-схема аудио кодера 1000 согласно варианту реализации изобретения. Аудио кодер 1000 согласно фиг. 10 подобен аудио кодеру 700 в соответствии с фиг. 7, поэтому идентичные сигналы и средства обозначены одинаковыми

35 номерами ссылок на фиг. 7 и 10.

Аудио кодер 1000 настроен получать входную аудио информацию 710 и обеспечивать на ее основе кодированную аудио информацию 712. Аудио кодер 1000 включает энергосберегающий конвертер временной области в частотную область, который настроен обеспечивать представление в частотной области 722 на основе представления

40 входной аудио информации 710 во временной области таким образом, что аудио представление в частотной области 722 включает набор спектральных значений. Аудио кодер 1000 также включает арифметический кодер 1030, настроенный кодировать спектральное значение (из набора спектральных значений, формирующих аудио представление в частотной области 722) или ее ранее обработанную версию при помощи

45 кодового слова переменной длины для того, чтобы получить кодированную аудио информацию 712 (которая может включать, например, множество кодовых слов переменной длины).

Арифметический кодер 1030 настроен отображать спектральное значение или

множество спектральных значений, либо значение наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода (т.е. на кодовое слово переменной длины) в зависимости от состояния контекста.

Арифметический кодер 1030 настроен выбирать правило отображения, описывающее  
5 отображение спектрального значения или множества спектральных значений, либо наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода в зависимости от состояния контекста. Арифметический кодер настроен определять текущее состояние контекста в зависимости от множества ранее кодированных (предпочтительно, но не обязательно смежных)  
10 спектральных значений. Для этой цели арифметический кодер настроен модифицировать числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее одному или более ранее кодированному спектральному значению (например, для того, чтобы выбрать соответствующее правило отображения), в зависимости от значения поддиапазона контекста для того, чтобы  
15 получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, соответствующее одному или более кодируемому спектральному значению (например, для того, чтобы выбрать соответствующее правило отображения).

Как можно видеть, отображение спектрального значения или множества  
20 спектральных значений, либо наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода может осуществляться при помощи кодирования спектрального значения 740 при использовании правила отображения, описанного в информации правила отображения 742. Трекер состояния 750 настроен отслеживать состояние контекста. Трекер состояния 750 настроен  
25 модифицировать числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее кодированию одного или более кодируемого спектрального значения, в зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, соответствующее  
30 кодированию одного или более кодируемого спектрального значения. Модификация числового представления числового значения предыдущего контекста может осуществляться, например, при помощи модификатора числового представления 1052, который получает числовое значение предыдущего контекста или одно или более значение поддиапазона контекста и обеспечивает числовое значение текущего контекста.  
35 Соответственно, трекер состояния 1050 обеспечивает информацию 754, описывающую состояние текущего контекста, например, в форме числового значения текущего контекста. Селектор правила отображения 1060 может выбирать правило отображения, например, сводную таблицу частот, описывающую отображение значения или множества спектральных значений, либо наиболее значимой битовой плоскости спектрального  
40 значения или множества спектральных значений на значение кода. Соответственно, селектор правила отображения 1060 обеспечивает информацию правила отображения 742 для спектрального кодирования 740.

Необходимо отметить, что в некоторых вариантах реализации изобретения трекер состояния 1050 может быть идентичен трекеру состояния 750 или трекеру состояния  
45 826. Также необходимо отметить, что селектор правила отображения 1060 в некоторых вариантах реализации изобретения может быть идентичен селектору правила отображения 760 или селектору правила отображения 828.

Суммируя вышесказанное, аудио кодер 1000 выполняет арифметическое кодирование

аудио представления в частотной области, которое обеспечивается конвертером временной области в частотную область. Арифметическое кодирование зависит от контекста, поэтому выбирается правило отображения (например, сводная таблица частот) в зависимости от ранее кодированных спектральных значений. Соответственно, спектральные значения, смежные по времени и/или по частоте (или по меньшей мере в пределах заданного окружения) по отношению друг к другу и/или к кодируемому спектральному значению (т.е. спектральные значения в пределах заданного окружения кодируемого спектрального значения) рассматривается в арифметическом кодировании, чтобы адаптироваться к распределению возможности, которое оценивается в арифметическом кодировании.

При определении числового значения текущего контекста модифицируется числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее одному или более ранее кодированному спектральному значению, в зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее одному или более кодируемому спектральному значению. Такой подход позволяет избежать полного повторного вычисления числового значения текущего контекста, в то время как при обычном подходе полное повторное вычисление требует значительных ресурсных затрат. Существует большое количество возможностей для модификации числового представления числового значения предыдущего контекста, включая комбинацию повторного масштабирования числового представления числового значения предыдущего контекста, добавление значения поддиапазона контекста или значения, извлеченного из него, к числовому представлению числового значения предыдущего контекста или к обработанному числовому представлению числового значения предыдущего контекста, замена части числового представления (а не всего числового представления) числового значения предыдущего контекста в зависимости от значения поддиапазона контекста и т.д. Таким образом, обычно числовое представление числового значения предыдущего контекста формируется на основе числового представления числового значения предыдущего контекста, а также на основе по меньшей мере одного значения поддиапазона контекста, при этом обычно выполняется комбинация операций, чтобы объединить числовое значение предыдущего контекста со значением поддиапазона контекста, таких как, например, операция сложения, операция вычитания, операция умножения, операция деления, булева операция AND, булева операция OR, булева операция NAND, булева операция NOR, булева операция отрицания, операция дополнения или операция перемещения. Соответственно, по меньшей мере часть числового представления числового значения предыдущего контекста обычно остается неизменной (за исключением опционального перемещения на другую позицию) при извлечении числового значения текущего контекста из числового значения предыдущего контекста. Наоборот, другие части числового представления числового значения предыдущего контекста изменяются в зависимости от одного или более значений поддиапазона контекста. Таким образом, числовое значение текущего контекста может быть получено при сравнительно небольших вычислительных затратах, при этом избегая полного повторного вычисления числового значения текущего контекста.

Таким образом, может быть получено имеющее смысл числовое значение текущего контекста, что подходит для селектора правила отображения 1060.

Следовательно, возможно добиться эффективного кодирования, оставляя вычисление

контекста достаточно простым.

#### 5. Аудио декодер в соответствии с фиг. 11

На фиг. 11 показана блок-схема аудио декодера. Аудио декодер 1100 подобен аудио декодеру 800 в соответствии с фиг. 8, поэтому идентичные сигналы, средства и функции обозначены одинаковыми номерами ссылок.

Аудио декодер 1100 настроен получать кодированную аудио информацию 810 и обеспечивать на ее основе декодированную аудио информацию 812. Аудио декодер 1100 включает арифметический декодер 1120, настроенный обеспечивать множество декодированных спектральных значений 822 на основе арифметически кодированного представления 821 спектральных значений. Аудио декодер 1100 также включает конвертер частотной области во временную область 830, который настроен получать декодированные спектральные значения 822 и обеспечивать аудио представление во временной области 812, которое может составлять декодированную аудио информацию, используя декодированные спектральные значения 822, для того, чтобы получить декодированную аудио информацию 812.

Арифметический декодер 1120 включает определитель спектрального значения 824, который настроен отображать значение кода арифметически кодированного представления 821 спектральных значений на код символа, представляющий одно или более декодированное спектральное значение или по меньшей мере часть (например, наиболее значимой битовой плоскости) одного или более декодированных спектральных значений. Определитель спектрального значения 824 настроен выполнять отображение в зависимости от правила отображения, которое может быть описано с помощью информации правила отображения 828а.

Информация правила отображения 828а может включать, например, значение индекса правила отображения или выборочный набор записей сводной таблиц частот.

Арифметический декодер 1120 настроен выбирать правило отображения (например, сводную таблицу частот), которое описывает отображение значения кода (описанного с помощью арифметически кодированного представления 821 спектральных значений) на код символа (которое описывает одно или более спектральных значений) в зависимости от состояния контекста, данное состояние контекста может быть описано с помощью информации о состоянии контекста 1126а. Информация о состоянии контекста 1126а может иметь форму числового значения текущего контекста.

Арифметический декодер 1120 настроен определять состояние текущего контекста в зависимости от множества ранее декодированных спектральных значений 822. Для этой цели может быть использован трекер состояния 1126, который получает информацию, описывающую ранее декодированные спектральные значения.

Арифметический декодер настроен модифицировать числовое представление числового значения предыдущего контекста, которое описывает состояние контекста, соответствующее одному или более ранее декодированным спектральным значениям, в зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, соответствующее одному или более декодируемым спектральным значениям. Модификация числового представления числового значения предыдущего контекста может осуществляться, например, с помощью модификатора числового представления 1127, который является частью трекера состояния 1126. Соответственно, может быть получена информация о состоянии текущего контекста 1126а, например, в форме числового значения текущего контекста. Выбор правила отображения может осуществляться с помощью селектора правила отображения 1128, который извлекает

информацию правила отображения 828a из информации о состоянии текущего контекста 1126a и обеспечивает информацию правила отображения 828a для определителя спектрального значения 824.

5 Что касается функциональности декодера аудио сигнала 1100, необходимо отметить, что арифметический декодер 1120 настроен выбирать правило отображения (например, сводную таблицу частот), которое обычно хорошо адаптируется к декодируемому спектральному значению, т.к. правило отображения выбирается в зависимости от состояния текущего контекста, который, в свою очередь, определяется в зависимости от множества ранее декодированных спектральных значений. Соответственно, можно  
10 применять статистические зависимости между декодируемыми смежными спектральными значениями.

Кроме этого, при модификации числового представления числового значения предыдущего контекста, описывающего состояние контекста, соответствующего декодированию одного или более ранее декодированных спектральных значений, в  
15 зависимости от значения поддиапазона контекста для того, чтобы получить числовое представление числового значения текущего контекста, описывающего состояние контекста, соответствующего декодированию одного или более декодируемых спектральных значений, возможно получить значимую информацию о состоянии текущего контекста, которая подходит для отображения значения индекса правила  
20 отображения, при сравнительно низких затратах на вычисление. При обработке по меньшей мере части числового представления числового значения предыдущего контекста (возможно бит-смещенная или масштабированная версия) в ходе обновления другой части числового представления числового значения предыдущего контекста в зависимости от значений поддиапазона контекста, которые не были учтены в числовом  
25 значении предыдущего контекста, но которые должны быть учтены в числовом значении текущего контекста, количество операций по извлечению числового значения текущего контекста может оставаться незначительным. Также возможно применять тот факт, что контексты, используемые для декодирования смежных спектральных значений обычно подобны или коррелируют друг с другом. Например, контекст для  
30 декодирования первого спектрального значения (или первого множества спектральных значений) зависит от первого набора ранее декодированных спектральных значений. Контекст для декодирования второго спектрального значения (или второго множества спектральных значений), который является смежным с первым спектральным значением (или первым набором спектральных значений) может включать второй набор ранее  
35 декодированных спектральных значений. Так как предполагается, что первое спектральное значение и второе спектральное значение являются смежными (например, в отношении соответствующих частот), первый набор спектральных значений, который определяет контекст для кодирования первого спектрального значения, может включать некоторое пересечение со вторым набором спектральных значений, который определяет  
40 контекст для декодирования второго спектрального значения. Соответственно, становится понятным, что состояние контекста для декодирования второго спектрального значения включает некоторую корреляцию с состоянием контекста для декодирования первого спектрального значения. Используя такие корреляции, можно достичь эффективности вычисления при извлечении контекста, т.е. при извлечении  
45 числового значения текущего контекста. Было обнаружено, что можно эффективно применять корреляцию между состояниями контекста для декодирования смежных спектральных значений (например, между состоянием контекста, описанного с помощью числового значения предыдущего контекста, и состоянием контекста, описанного с

помощью числового значения текущего контекста), модифицируя только те части числового значения предыдущего контекста, которые зависят от значений поддиапазона контекста, не рассматриваемых для извлечения числового значения состояния предыдущего контекста, а также извлекая числовое значение текущего контекста из

5 числового значения предыдущего контекста.

Таким образом, описанная здесь концепция позволяет обеспечить высокую эффективность вычислений при извлечении числового значения текущего контекста.

Дальнейшие детали будут рассмотрены ниже.

6. Аудио кодер в соответствии с фиг. 12

10 На фиг. 12 показана блок-схема аудио кодера согласно варианту реализации изобретения. Аудио кодер 1200 согласно фиг. 12 подобен аудио кодеру 700 в соответствии с фиг. 7, поэтому идентичные сигналы, средства и функции обозначены одинаковыми номерами ссылок.

Аудио кодер 1200 настроен получать входную аудио информацию 710 и обеспечивать

15 на ее основе кодированную аудио информацию 712. Аудио кодер 1200 включает энергосберегающий конвертер временной области в частотную область 720, который настроен обеспечивать аудио представление в частотной области 722 на основе представления входной аудио информации 710 во временной области таким образом, что аудио представление в частотной области 722 включает набор спектральных

20 значений. Аудио кодер 1200 также включает арифметический кодер 1230, настроенный кодировать спектральное значение (из набора спектральных значений, формирующих аудио представление в частотной области 722) или множество спектральных значений, или их ранее обработанную версию при помощи кодового слова переменной длины для того, чтобы получить кодированную аудио информацию 712 (которая может

25 включать, например, множество кодовых слов переменной длины).

Арифметический кодер 1230 настроен отображать спектральное значение или множество спектральных значений, либо значение наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода (т.е. на кодовое слово переменной длины) в зависимости от состояния контекста.

30 Арифметический кодер 1230 настроен выбирать правило отображения, описывающее отображение спектрального значения или множества спектральных значений, либо наиболее значимой битовой плоскости спектрального значения или множества спектральных значений, на значение кода в зависимости от состояния контекста. Арифметический кодер настроен определять текущее состояние контекста в зависимости

35 от множества ранее кодированных (предпочтительно, но не обязательно смежных) спектральных значений. Для этой цели арифметический кодер настроен получать множество значений поддиапазона контекста на основе ранее кодированных спектральных значений, сохранять указанные значения поддиапазона контекста и извлекать числовое значение текущего контекста, соответствующее одному или более

40 кодируемому спектральному значению в зависимости от сохраненных значений поддиапазона контекста. Кроме этого, арифметический кодер настроен вычислять норму вектора, формируемого множеством ранее кодированных спектральных значений, для того, чтобы получить общее значение поддиапазона контекста, соответствующее множеству ранее кодированных спектральных значений.

45 Как можно видеть, отображение спектрального значения или множества спектральных значений, либо наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода может осуществляться при помощи кодирования спектрального значения 740 при использовании правила



отображения, описанного в информации правила отображения 742. Трекер состояния 1250 настроен отслеживать состояние контекста и может включать блок вычисления значения поддиапазона контекста 1252 для того, чтобы вычислять норму вектора, которая формируется множеством ранее кодированных спектральных значений, с целью получить общие значения поддиапазона контекста, соответствующие множеству ранее кодированных спектральных значений. Трекер состояния 1250 также предпочтительно настроен определять состояние текущего контекста в зависимости от результата указанного вычисления значения поддиапазона контекста, которое осуществляется блоком вычисления значения поддиапазона контекста 1252.

Соответственно, трекер состояния 1250 предоставляет информацию 1254, которая описывает состояние текущего контекста. Селектор правила отображения 1260 может выбирать правило отображения, например, сводную таблицу частот, описывающую отображение спектрального значения или множества спектральных значений, либо наиболее значимой битовой плоскости спектрального значения или множества спектральных значений на значение кода. Соответственно, селектор правила отображения 1260 обеспечивает информацию правила отображения 742 для спектрального кодирования 740.

Суммируя вышесказанное, аудио кодер 1200 выполняет арифметическое кодирование аудио представления в частотной области, которое обеспечивается конвертером временной области в частотную область. Арифметическое кодирование зависит от контекста, поэтому выбирается правило отображения (например, сводная таблица частот) в зависимости от ранее кодированных спектральных значений. Соответственно, спектральные значения, смежные по времени и/или по частоте (или по меньшей мере в пределах заданного окружения) по отношению друг к другу и/или к кодируемому спектральному значению (т.е. спектральные значения в пределах заданного окружения кодируемого спектрального значения) рассматривается в арифметическом кодировании, чтобы адаптировать распределение возможности, которое оценивается в арифметическом кодировании.

Чтобы обеспечить числовое значение текущего контекста, определяется значение поддиапазона контекста, соответствующее множеству ранее кодированных спектральных значений, на основе вычисления нормы вектора, формируемого множеством ранее кодированных спектральных значений. Результат определения числового значения текущего контекста применяется при выборе состояния текущего контекста, т.е. при выборе правила отображения.

При вычислении нормы вектора, формируемого множеством ранее кодированных спектральных значений, может быть получена значимая информация, описывающая часть контекста одного или более кодируемых спектральных значений, при этом норма вектора ранее кодированных спектральных значений контекстной информации, необходимой для дальнейшего использования при извлечении числового значения текущего контекста, может оставаться незначительным, если применяется рассматриваемый выше подход для вычисления значений поддиапазона контекста. Было обнаружено, что норма вектора ранее кодированных спектральных значений обычно включает наиболее значимую информацию относительно состояния контекста. И наоборот, было обнаружено, что знак указанных ранее кодированных спектральных значений обычно имеет побочное воздействие на состояние контекста, поэтому будет рациональным пренебречь знаком ранее декодированных спектральных значений для того, чтобы уменьшить количество информации, которая сохраняется для дальнейшего использования. Было обнаружено, что вычисление нормы вектора ранее кодированных

спектральных значений является рациональным подходом для извлечения значения поддиапазона контекста, так как эффект усреднения, обычно получаемый при вычислении нормы, значительно не затрагивает наиболее важную информацию о состоянии контекста. Таким образом, вычисление значения поддиапазона контекста, выполняемое блоком вычисления значения поддиапазона контекста 1252 позволяет обеспечить компактную информацию о поддиапазоне контекста для хранения и дальнейшего использования, при этом наиболее значимая информация о состоянии контекста сохраняется, несмотря на уменьшение количество информации.

Соответственно, можно добиться эффективного кодирования входной аудио информации 710, при этом затраты на вычисление и количество данных, сохраняемых арифметическим кодером 1230, остаются незначительными.

#### 7. Аудио декодер в соответствии с фиг. 13

На фиг. 13 показана блок-схема аудио декодера 1300. Аудио декодер 1300 подобен аудио декодеру 800 в соответствии с фиг. 8 и аудио декодеру 1100 в соответствии с фиг. 11, поэтому идентичные сигналы, средства и функции обозначены одинаковыми номерами ссылок.

Аудио кодер 1300 настроен получать кодированную аудио информацию 810 и обеспечивать на ее основе декодированную аудио информацию 812. Аудио декодер 1300 включает арифметический декодер 1320, настроенный обеспечивать множество декодированных спектральных значений 822 на основе арифметически кодированного представления 821 спектральных значений. Аудио декодер 1300 также включает конвертер частотной области во временную область 830, который настроен получать декодированные спектральные значения 822 и обеспечивать аудио представление во временной области 812, которое может составлять декодированную аудио информацию, используя декодированные спектральные значения 822, для того, чтобы получить декодированную аудио информацию 812.

Арифметический декодер 1320 включает определитель спектрального значения 824, который настроен отображать значение кода арифметически кодированного представления 821 спектральных значений на код символа, представляющий одно или более декодированное спектральное значение или по меньшей мере часть (например, наиболее значимую битовую плоскость) одного или более декодированных спектральных значений. Определитель спектрального значения 824 настроен выполнять отображение в зависимости от правила отображения, которое может быть описано с помощью информации правила отображения 828а. Информация правила отображения 828а может включать, например, значение индекса правила отображения или выборочный набор записей сводной таблиц частот.

Арифметический декодер 1320 настроен выбирать правило отображения (например, сводную таблицу частот), которое описывает отображение значения кода (описанного с помощью арифметически кодированного представления 821 спектральных значений) на код символа (которое описывает одно или более спектральное значение) в зависимости от состояния контекста (которое может быть описано при помощи информации о состоянии контекста 1326а). Арифметический декодер 1320 настроен определять состояние текущего контекста в зависимости от множества ранее декодированных спектральных значений 822. Для этой цели может быть использован трекер состояния 1326, который получает информацию, описывающую ранее декодированные спектральные значения. Арифметический декодер также настроен получать множество значений поддиапазона контекста на основе ранее декодированных спектральных значений и сохранять указанные значения поддиапазона контекста.

Арифметический декодер настроен извлекать числовое значение текущего контекста, соответствующее одному или более декодируемому спектральному значению в зависимости от сохраненных значений поддиапазона контекста. Арифметический декодер 1320 настроен вычислять норму вектора, формируемого множеством ранее декодированных спектральных значений, для того, чтобы получить общее значение поддиапазона контекста, соответствующее множеству ранее декодированных спектральных значений.

Вычисление нормы вектора, формируемого множеством ранее кодированных спектральных значений, для получения общего значения поддиапазона контекста, соответствующего множеству ранее декодированных спектральных значений, может выполняться, например, с помощью блока вычисления значения поддиапазона контекста 1327, который является частью трекера состояния 1326. Соответственно, может быть получена информация о состоянии текущего контекста 1326а на основе значений поддиапазона контекста, при этом трекер состояния 1326 предпочтительно обеспечивает числовое значение текущего контекста, соответствующее одному или более декодируемому спектральному значению, в зависимости от сохраненных значений поддиапазона контекста. Выбор правила отображения может осуществляться с помощью селектора правила отображения 1328, который извлекает информацию правила отображения 828а из информации о состоянии текущего контекста 1326а и обеспечивает информацию правила отображения 828а для определителя спектрального значения 824.

Что касается функциональности декодера аудио сигнала 1300, необходимо отметить, что арифметический декодер 1320 настроен выбирать правило отображения (например, сводную таблицу частот), которое обычно хорошо адаптируется к декодируемому спектральному значению, т.к. правило отображения выбирается в зависимости от состояния текущего контекста, который, в свою очередь, определяется в зависимости от множества ранее декодированных спектральных значений. Соответственно, можно применять статистические зависимости между декодируемыми смежными спектральными значениями.

Было обнаружено, что в плане использования памяти эффективно хранить значения поддиапазона контекста, которые основаны на вычислении нормы вектора, формируемого множеством ранее декодированных спектральных значений, для их дальнейшего применения при определении значения контекста. Также было обнаружено, что такие значения поддиапазона контекста содержат наиболее значимую информацию о контексте. Соответственно, концепция, на основе которой функционирует трекер состояния 1326, обеспечивает компромисс между эффективностью кодирования и затратами на вычисления и хранение данных.

Дальнейшие детали будут рассмотрены ниже.

#### 8. Аудио кодер в соответствии с фиг. 1

Далее будет описан аудио кодер в соответствии с вариантом реализации настоящего изобретения. Фиг. 1 показывает блок-схему такого аудио кодера 100.

Аудио кодер 100 настроен на получение входной аудио информации 110 и на предоставление на ее основе битового потока 112, который представляет собой кодированную аудио информацию. Аудио декодер 100 может дополнительно включать препроцессор 120, который настроен на получение входной аудио информации 110 и предоставление на ее основе предварительно обработанной входной аудио информации 110а. на фиг. Аудио кодер 100 также включает энергосберегающий конвертер сигнала из временной области в частотную область 130, который также обозначается как

конвертер сигнала. Конвертер сигнала 130 настроен на получение входной аудио информации 110, 110а и предоставление на ее основе аудио информации 132 в частотной области, которая предпочтительно имеет вид набора спектральных значений. Например, трансформер сигнала 130 может быть настроен для получения фрейма входной аудио информации 110, 110а (например, блок образцов временной области) и для предоставления набора спектральных значений, представляющих аудио содержание соответствующего аудио фрейма. Кроме того, трансформер сигнала 130 может быть настроен на получение множества последующих, пересекающихся или непересекающихся аудио фреймов входной аудио информации 110, 110а и предоставление на ее основе аудио представления во временной и частотной области, которое состоит из последовательности последующих наборов спектральных значений, один набор спектральных значений связан с каждым фреймом.

Энергосберегающий трансформер сигнала из временной области в частотную область 130 может включать в себя энергосберегающий банк фильтров, который обеспечивает спектральные значения, соответствующие различным, пересекающимся или непересекающимся, частотным диапазонам. Например, трансформер сигнала 130 может включать в себя оконный MDCT трансформер 130а, который настроен разделять на окна входную аудио информацию 110, 110а (или ее фрейм) с помощью окна преобразования и выполнения модифицированного дискретного косинус-преобразования разделенной на окна входной аудио информации 110, 110а (или разделенного на окна фрейма). Таким образом, аудио представление в частотной области 132 может включать в себя набор, например, 1024 спектральных значений в виде MDCT коэффициентов, соответствующих фрему входной аудио информации.

Аудио декодер 100 может дополнительно включать спектральный постпроцессор 140, который настроен на получение аудио представления в частотной области 132 и предоставление на ее основе пост обработанного аудио представления в частотной области 142. Спектральный постпроцессор 140 может, например, быть настроен на выполнение временного ограничения шума и/или долгосрочного прогноза и/или любой другой спектральной пост-обработки, известной в данной области. Аудио кодер дополнительно содержит, по желанию, скейлер/квантователь 150, который настроен на получение в частотной области аудио представления 132 или версию его пост-обработки 142 и для обеспечения масштабированного и квантованного аудио представления в частотной области 152.

Аудио кодер 100 далее содержит, по желанию, психо-акустическую модель процессора 160, который настроен на получение входной аудио информации 110 (или пост-обработанной версии 110а) и представление на ее основе дополнительной контрольной информации, которая может быть использована для управления энергосберегающим трансформером сигнала из временной области в частотную область 130 для управления дополнительным спектральным пост-процессором 140 и/или для контроля за дополнительным скейлером/квантователем 150. Например, психо-акустическая модель процессора 160 может быть сконфигурирована для анализа входной аудио информации, чтобы определить, какие компоненты входной аудио информации 110, 110а особенно важны для человеческого восприятия аудио содержания и какие компоненты входной аудио информации 110, 110а менее важны для восприятия аудио содержания. Таким образом, психо-акустическая модель процессора 160 может обеспечить контрольную информацию, которая используется аудио кодером 100 для регулировки масштабирования аудио представления в частотной области 132, 142 скейлером/квантователем 150 и/или разрешения квантования, которое применяется скейлером/

квантователем 150. Следовательно, важные для восприятия группы масштабных коэффициентов (т.е. группы смежных спектральных значений, которые являются особенно важными для человеческого восприятия аудио содержания) масштабируются с большим коэффициентом масштабирования и квантуются со сравнительно высоким разрешением, в то время как менее важные для восприятия группы масштабных коэффициентов (т.е. группы смежных спектральных значений) масштабируются со сравнительно меньшим коэффициентом масштабирования и квантуются со сравнительно низким разрешением квантования. Таким образом, масштабированные спектральные значения частот более важных для восприятия, как правило, значительно больше, чем спектральные значения частот менее важных для восприятия.

Аудио кодер также включает в себя арифметический кодер 170, который настроен на получение масштабированной и квантованной версии 152 аудио представления в частотной области 132 (или, наоборот, пост-обработанной версии 142 аудио представления в частотной области 132, или даже само аудио представление в частотной области 132), а также на обеспечение арифметической информации кодового слова 172а на ее основе таким образом, что арифметическая информация кодового слова представляет аудио представление в частотной области 152.

Аудио кодер 100 также включает в себя форматтер полезной нагрузки битового потока 190, который настроен на получение арифметической информации кодового слова 172а. Форматтер полезной нагрузки битового потока 190 также обычно настроен на получение дополнительной информации, как, например, информации коэффициента масштабирования, описывающей, какие коэффициенты масштабирования были применены скейлером/квантователем 150. Кроме того, форматтер полезной нагрузки битового потока 190 может быть настроен на получение другой управляющей информации. Форматтер полезной нагрузки битового потока 190 настроен на обеспечение битового потока 112 на основе полученной информации путем сборки битового потока в соответствии с желаемым синтаксисом потока, который будет обсуждаться ниже.

Далее будут рассмотрены подробности, касающиеся арифметического кодера 170. Арифметический кодер 170 настроен на получение множества пост-обработанных, масштабированных и квантованных спектральных значений аудио представления в частотной области 132. Арифметический кодер включает в себя экстрактор наиболее значимой битовой плоскости 174, который настроен на извлечение наиболее значимой битовой плоскости  $m$  спектрального значения или даже двух спектральных значений. Следует отметить, что наиболее значимая битовая плоскость может содержать один или более битов (например, два или три бита), которые являются наиболее значимыми битами спектрального значения. Таким образом, экстрактор наиболее значимой битовой плоскости 174 обеспечивает значение наиболее значимой битовой плоскости 176 спектрального значения.

Однако экстрактор наиболее значимой битовой плоскости 174 может обеспечивать комбинированное наиболее значимое значение битовой плоскости  $m$ , которое объединяет наиболее значимые битовые плоскости множества спектральных значений (например, спектральных значений  $a$  и  $b$ ). Наиболее значимая битовая плоскость спектрального значения  $a$  обозначается как  $m$ . В качестве альтернатив комбинированное значение наиболее значимой битовой плоскости множества спектральных значений  $a$ ,  $b$  также обозначается как  $m$ .

Арифметический кодер 170 также включает первый определитель кодового слова 180, который настроен определять арифметическое кодовое слово  $acod\_m [pki][m]$ ,

представляющее значение наиболее значимой битовой плоскости - значение  $m$ . По желанию, определитель кодового слова 180 может также предоставить одно или более управляющее кодовое слово (также обозначенные здесь с "ARITH\_ESCAPE") с указанием, например, сколько доступно менее значимых битовых плоскостей (и, следовательно, указывающих числовой вес наиболее значимой битовой плоскости).  
Первый определитель кодового слова 180 может быть сконфигурирован для обеспечения кодового слова, соответствующего значению наиболее значимой битовой плоскости  $m$  с помощью выбранной сводной таблицы частот, имеющей (или которая ссылается на) индекс сводной таблицы частоты  $pk_i$ .

Для того чтобы определить, какую сводную таблицу частот надо выбрать, арифметический кодер предпочтительно включает трекер состояния 182, который настроен на отслеживание состояния арифметического кодера, например, с помощью наблюдения за тем, какие спектральные значения были кодированы ранее. Трекер состояния 182, следовательно, дает информацию о состоянии 184, например, значение состояния обозначается "s" или "t". Арифметический кодер 170 также включает селектор сводной таблицы частот 186, который настроен на получение информации о состоянии 184 и предоставление информации 188, описывающей выбранную сводную таблицу частот для определителя кодового слова 180. Например, селектор сводной таблицы частот 186 может предоставить индекс сводной таблицы частот „ $pk_i$ “, описывающий, какая сводная таблица частот из набора из 96 сводных таблиц частот выбрана для использования определителем кодового слова. Кроме того, селектор сводной таблицы частот 186 может предоставить всю выбранную сводную таблицу частот или часть таблицы для определителя кодового слова. Таким образом, определитель кодового слова 180 может использовать выбранную сводную таблицу частот или часть таблицы для предоставления кодового слова  $acod\_m[pk_i][m]$  значения наиболее значимой битовой плоскости  $in$ , так что фактическое кодовое слово  $acod\_m[pk_i][m]$ , кодирующее значение наиболее значимой битовой плоскости  $m$  зависит от значения  $m$  и индекса сводной таблицы частот  $pk_i$ , и, следовательно, от информации о текущем состоянии 184. Более подробная информация о процессе кодирования и формате полученного кодового слова будет рассмотрена ниже.

Необходимо отметить, что в некоторых вариантах реализации изобретения трекер состояния 182 может быть идентичен по функции трекеру состояния 750, трекеру состояния 1050 или трекеру состояния 1250. Селектор сводной таблицы частот 186 в некоторых вариантах реализации изобретения может быть идентичен по функции селектору правила отображения 760, селектору правила отображения 1060 или селектору правила отображения 1260. Кроме этого, первый определитель кодового слова 180 в некоторых вариантах реализации изобретения может быть идентичен по функции блоку кодирования спектрального значения 740.

Арифметический кодер 170 включает экстрактор менее значимой битовой плоскости 189а, который настроен на извлечение одной или более менее значимой битовой плоскости из масштабированного и квантованного аудио представления в частотной области 152, если одно или более кодируемое спектральное значение превышает диапазон кодируемых значений, используя только наиболее значимую битовую плоскость. Менее значимые битовые плоскости могут включать в соответствии с необходимостью один или более битов. Соответственно, экстрактор менее значимой битовой плоскости 189а предоставляет информацию о менее значимой битовой плоскости 189б. Арифметический кодер 170 также включает второй определитель кодового слова 189с, который настроен на получение информации о менее значимой

битовой плоскости 189d и предоставление на ее основе 0, 1 или более кодовых слов "acod\_r", представляющих содержание 0, 1 или более менее значимых битовых плоскостей. Второй определитель кодового слова 189с может быть настроен на применение алгоритма арифметического кодирования или любой другой алгоритм кодирования для того, чтобы извлечь кодовые слова менее значимых битовых плоскостей "acod\_r" из информации о менее значимой битовой плоскости 189b.

Следует отметить, что количество менее значимых битовых плоскостей может варьироваться в зависимости от значения масштабированных и квантованных спектральных значений 152, так что менее значимые битовые плоскости могут вообще отсутствовать, если масштабированное и квантованное кодируемое спектральное значение сравнительно невелико, так, например, может быть одна менее значимая битовая плоскость, если кодируемое в настоящий момент масштабированное и квантованное спектральное значение является средним по размеру или, например, может быть более одной менее значимой битовой плоскости, если кодируемое масштабированное и квантованное спектральное значение является сравнительно большим.

Подводя итог вышесказанному, арифметический кодер 170 настроен на кодирование масштабированных и квантованных спектральных значений, которые описаны в информации 152, с помощью иерархического процесса кодирования. Наиболее значимая битовая плоскость (содержащая, например, один, два или три бита на спектральное значение) одного или более спектральных значений кодируется для получения арифметического кодового слова "acod\_m[*pk*][*m*]" значения *m* наиболее значимой битовой плоскости. Одна или более менее значимых битовых плоскостей (каждая из менее значимых битовых плоскостей включает, например, один, два или три бита) одного или более спектральных значений кодируются, чтобы получить одно или более кодовых слов "acod\_r". При кодировании наиболее значимой битовой плоскости значение *m* наиболее значимой битовой плоскости отображается на кодовое слово. acod\_m[*pk*][*m*]. Для этого 96 различных сводных таблиц частот доступны для кодирования значения *m* в зависимости от состояния арифметического кодера 170, т.е. в зависимости от ранее кодированных спектральных значений. Таким образом, получается кодовое слово "acod\_m[*pk*][*m*]" . Кроме того, одно или более кодовых слов "acod\_r" предусмотрены и включены в битовый поток, если присутствуют одна или более менее значимых битовых плоскостей.

#### Описание сброса

Аудио кодер 100 может быть дополнительно настроен принимать решение о том, можно ли достичь повышения битрейта путем сброса контекста, например, установив индекса состояния на значение по умолчанию. Таким образом, аудио кодер 100 может быть сконфигурирован для обеспечения информации сброса (например, под названием "arith\_reset\_flag"), указывающей, является ли контекст для арифметического кодирования сброшенным, а также указывающей, следует ли сбросить контекст для арифметического декодирования в соответствующем декодере.

Подробнее формат битового потока и применяемые сводные таблицы частот будут рассмотрены ниже.

#### 9. Аудио декодер в соответствии с фиг. 2

Далее будет описан аудио декодер в соответствии с вариантом реализации настоящего изобретения. Фиг. 2 показывает блок-схему такого аудио декодера 200.

Аудио декодер 200 настроен на получение битового потока 210, который представляет кодированную аудио информацию и который может быть идентичен битовому потоку

112, который предоставляет кодер 100. Аудио декодер 200 обеспечивает декодированную аудио информацию 212 на основе битового потока 210.

Аудио декодер 200 включает дополнительный де-форматтер полезной нагрузки битового потока 220, который настроен на получение битового потока 210 и извлечение из битового потока 210 кодированного аудио представления в частотной области 222. Например, де-форматтер полезной нагрузки битового потока 220 может быть настроен на извлечение из битового потока 210 арифметически кодированных спектральных данных, таких как, например, арифметическое кодовое слово "acod\_m [rki][m]", представляющее значение наиболее значимой битовой плоскости m спектрального значения a или множества спектральных значений a, b, а также кодовое слово "acod\_r", представляющее содержание менее значимой битовой плоскости спектрального значения a или множества спектральных значений a, b в аудио представлении в частотной области. Таким образом, кодированное аудио представление в частотной области 222 составляет (или включает) арифметически кодированное представление спектральных значений. Де-форматтер полезной нагрузки битового потока 220 дополнительно настроен на извлечение из битового потока дополнительной информации управления, которая не показана на фиг. 2. Кроме того, де-форматтер полезной нагрузки битового потока дополнительно настроен на извлечение из битового потока 210 информации сброса состояния 224, которая также обозначается как арифметический флаг сброса или "arith\_reset\_flag".

Аудио декодер 200 включает арифметический декодер 230, который также обозначается как "спектральный бесшумный декодер". Арифметический декодер 230 настроен на прием кодированного аудио представления в частотной области 220 и, при необходимости, информации о сбросе состояния 224. Арифметический декодер 230 также настроен на предоставление декодированного аудио представления в частотной области 232, которое может включать декодированное представление спектральных значений. Например, декодированное аудио представление в частотной области 232 может содержать декодированное представление спектральных значений, которые описаны в кодированном аудио представлении в частотной области 220.

Аудио декодер 200 также включает дополнительный обратный квантователь/ресейлер 240, который настроен на получение декодированного аудио представления в частотной области 232 и предоставление на его основе обратно квантованного и ре-масштабированного аудио представления в частотной области 242.

Аудио декодер 200 также дополнительно может включать спектральный пред-процессор 250, который настроен на получение обратно квантованного и ре-масштабированного аудио представления в частотной области 242 и предоставление на его основе предварительно обработанной версии 252 обратно квантованного и ре-масштабированного аудио представления в частотной области 242. Аудио кодер 200 также включает в себя трансформер сигнала из частотной области во временную область 260, который также обозначается как конвертер сигнала. Трансформер сигнала 260 настроен на прием предварительно обработанной версии 252 обратно квантованного и ре-масштабированного аудио представления в частотной области 242 (или, наоборот, обратно квантованного и ре-масштабированного аудио представления в частотной области 242 или декодированного аудио представления в частотной области 232) и предоставление на его основе аудио информации представления во временной области 262. Трансформер сигнала из частотной области во временную область 260 может, например, включать трансформер для выполнения обратного модифицированного дискретного косинус-преобразования (IMDCT) и соответствующего



разделения на окна (а также других вспомогательных функций, как, например, перекрытие-и-добавление).

Аудио декодер 200 может дополнительно содержать пост-процессор временной области 270, который настроен на получение представления аудио информации во временной области 262 и получение декодированной аудио информации 212 с помощью пост-обработки во временной области. Однако, если пост-обработка отсутствует, представление во временной области 262 может быть идентичным декодированной аудио информации 212.

Следует отметить, что обратный квантователь/рескейлер 240, спектральный пред-процессор 250, трансформер сигнала из частотной области во временную область 260 и пост-процессор во временной области 270 управляются в зависимости от управляющей информации, которая извлекается из битового потока 210 с помощью де-форматтера полезной нагрузки битового потока 220.

Подводя итог общей функциональности аудио декодера 200, декодированное аудио представление в частотной области 232, например, набор спектральных значений, соответствующих аудио фрейму кодированной аудио информации, могут быть получены на основе кодированного представления в частотной области 222 с помощью арифметического декодера 230. Следовательно, набор, например, 1024 спектральных значений, которые могут быть MDCT коэффициентами, обратно квантованы, ре-масштабированы и предварительно обработаны. Соответственно, получается обратно квантованный, ре-масштабированный и спектрально предварительно обработанный набор спектральных значений (например, 1024 MDCT коэффициенты). Далее, представление во временной области аудио фрейма извлекается из обратно квантованного, ре-масштабированного и спектрально предварительно обработанного набора значений в частотной области (например, MDCT коэффициенты). Соответственно, получается представление аудио фрейма во временной области. Представление во временной области данного аудио фрейма может быть объединено с представлениями во временной области предыдущих и/или последующих аудио фреймов. Например, перекрытие-и-добавление между представлениями во временной области последующих аудио фреймов может быть выполнено для того, чтобы сгладить переходы между представлениями во временной области смежных аудио фреймов, а также с целью получения отмены сглаживания. Для получения дополнительной информации о реконструкции декодированной аудио информации 212 на основе декодированного аудио представления во временной области 232, делается ссылка, например, на международный стандарт ISO/IEC 14496-3, часть 3, раздел 4, где это детально обсуждается. Тем не менее, могут быть использованы другие более сложные схемы перекрытия и отмены наложения.

Далее будут рассмотрены подробности, касающиеся арифметического декодера 230. Арифметический декодер 230 включает определитель наиболее значимой битовой плоскости 284, который настроен на получение арифметического кодового слова `acod_m [pki][m]`, описывающего значение `m` наиболее значимой битовой плоскости. Определитель наиболее значимой битовой плоскости 284 может быть настроен на использование сводной таблицы частот из набора, содержащего множество из 96 сводных таблиц частот для извлечения значения `m` наиболее значимой битовой плоскости из арифметического кодового слова `"acod_m [pki][m]"`.

Определитель наиболее значимой битовой плоскости 284 настроен на извлечение значений 286 наиболее значимой битовой плоскости спектральных значений на основе кодового слова `acod_m`. Арифметический декодер 230 дополнительно включает

определитель наименее значимой битовой плоскости 288, который настроен на получение одного или более кодовых слов "acod\_r", представляющих одну или более менее значимую битовую плоскость спектрального значения. Соответственно, определитель наименее значимой битовой плоскости 288 настроен обеспечить декодированные значения 290 одной или более менее значимой битовой плоскости. Аудио декодер 200 также включает сумматор битовой плоскости 292, который настроен на получение декодированных значений 286 наиболее значимых битовых плоскостей спектральных значений и декодированных значений 290 одной или более менее значимых битовых плоскостей спектральных значений, если такие менее значимые битовые плоскости доступны для текущих спектральных значений. Соответственно, сумматор битовой плоскости 292 обеспечивает декодированные спектральные значения, которые являются частью декодированного аудио представления в частотной области 232. Естественно, арифметический декодер 230, как правило, настроены на предоставление множества спектральных значений для того, чтобы получить полный набор декодированных спектральных значений, соответствующих текущему фрейму аудио содержания.

Арифметический декодер 230 дополнительно включает селектор сводной таблицы частот 296, который настроен на выбор одной из 96 сводных таблиц частот в зависимости от индекса состояния 298, описывающего состояние арифметического декодера. Арифметический декодер 230 дополнительно включает трекер состояния 299, который настроен для отслеживания состояния арифметического декодера в зависимости от ранее декодированных спектральных значений. Информация о состоянии при необходимости может быть сброшена к информации состояния по умолчанию в ответ на информацию о сбросе состояния 224. Таким образом, селектор сводной таблицы частот 296 настроен для предоставления индекса (например,  $pki$ ), выбранной сводной таблицы частот или части самой выбранной сводной таблицы частот, для применения в декодировании значения  $m$  наиболее значимой битовой плоскости в зависимости от кодового слова "acod\_m".

Подводя итог функциональности аудио декодера 200, аудио декодер 200 настроен на получение битрейт эффективного кодированного аудио представления в частотной области 222 и получение декодированного аудио представления в частотной области на его основе. В арифметическом декодере 230, который используется для получения декодированного аудио представления в частотной области 232 на основе кодированного аудио представления в частотной области 222, вероятность различных комбинаций значений наиболее значимых битовых плоскостей смежных спектральных значений применяется арифметическим декодером 280, который настроен применять сводную таблицу частот. Другими словами, применяются статистические зависимости между спектральными значениями путем выбора различных сводных таблиц частот из набора, включающего 96 различных сводных таблиц частоты в зависимости от индекса состояния 298, который получается при наблюдении за ранее вычисленными декодированными спектральными значениями.

Необходимо отметить, что трекер состояния 299 может быть идентичен по функции трекеру состояния 826, трекеру состояния 1126 или трекеру состояния 11326. Селектор сводной таблицы частот 296 может быть идентичен по функции селектору правила отображения 828, селектору правила отображения 1128 или селектору правила отображения 1328. Определитель наиболее значимой битовой плоскости 284 может быть идентичен по функции определителю спектрального значения 824.

#### 10. Обзор за инструментов спектрального бесшумного кодирования

Далее будут рассмотрены подробности, касающиеся алгоритма кодирования и

декодирования, который выполняется, например, арифметическим кодером 170 и арифметическим декодером 230.

Основное внимание уделяется описанию алгоритма декодирования. Следует отметить, однако, что соответствующий алгоритм кодирования может быть выполнен в соответствии с объяснением алгоритма декодирования, в котором отображения 5 меняются на противоположные, при этом вычисления значения индекса правила отображения в значительной мере похожи. На стороне кодера кодируемые спектральные значения заменяются на декодируемые спектральные значения.

Следует отметить, что декодирование, которое будет обсуждаться далее, используется 10 для того, чтобы обеспечить так называемое "спектральное бесшумное кодирование" обычно пост-обработанных, масштабированных и квантованных спектральных значений. Спектральное бесшумное кодирование используется в концепции аудио кодирования/декодирования (или любой другой концепции кодирования/декодирования) для дальнейшего сокращения избыточности квантованного спектра, которое 15 выполняется, например, при помощи энергосберегающего трансформера из временной области в частотную область. Схема спектрального бесшумного кодирования, применяемая в вариантах реализации настоящего изобретения, основана на арифметическом кодировании в сочетании с динамически адаптированным контекстом.

В некоторых вариантах реализации изобретения схема спектрального бесшумного 20 кодирования основана на двойках, т.е. комбинации двух соседних спектральных коэффициентов. Каждая двойка разделена на знак, наиболее значимую 2-битовую плоскость, и оставшиеся менее значимые битовые плоскости. Бесшумное кодирование наиболее значимой 2-битовой плоскости  $m$  использует зависимые от контекста сводные таблицы частот, извлеченные из четырех ранее декодированных двоек. Бесшумное 25 кодирование использует квантованные спектральные значения и зависимые от контекста сводные таблицы частот, извлеченные из четырех ранее декодированных соседних двоек. Во внимание принимается смежное положение как по времени, так и по частоте, как показано на фиг. 4. Сводные таблицы частот (которые будут рассмотрены далее) затем используются арифметическим кодером для создания бинарного кода переменной 30 длины (а также арифметическим декодером для извлечения декодированных значений из бинарного кода переменной длины).

Например, арифметический кодер 170 производит бинарный код для данного набора символов и их соответствующих вероятностей (в зависимости от соответствующих 35 вероятностей). Бинарный код создается путем отображения интервала вероятности, в котором находится набор символов, на кодовое слово.

Бесшумное кодирование оставшейся менее значимой битовой плоскости  $g$  использует 40 одиночную таблицу сводных частот. Сводные частоты соответствуют, например, единому распределению символов, встречающихся в менее значимых битовых плоскостях, т.е. существует одинаковая вероятность появления 0 или 1 в менее значимых битовых плоскостях.

Далее будет дан еще один короткий обзор инструментов спектрального бесшумного кодирования. Спектральное бесшумное кодирование используется для дальнейшего сокращения избыточности квантованного спектра. Схема спектрального бесшумного кодирования основывается на арифметическом кодировании в сочетании с динамически 45 адаптированным контекстом. Бесшумное кодирование использует квантованные спектральные значения и контекстно-зависимые сводные таблицы частот, полученные, например, из четырех ранее декодированных соседних двоек спектральных значений. Здесь учитывается смежное расположение как во времени, так и по частоте, как показано

на фиг. 4. Сводные таблицы частот затем используются арифметическим кодером для создания бинарного кода переменной длины.

Арифметический кодер производит бинарный код для данного набора символов и их соответствующих вероятностей. Двоичный код образуется путем отображения интервала вероятности, в котором находится набор символов, на кодовое слово.

## 11. Процесс декодирования

### 11.1 Обзор процесса декодирования

Далее будет дан обзор процесса декодирования спектрального значения со ссылкой на фиг. 3, на которой показан псевдо-программный код процесса декодирования множества спектральных значений.

Процесс декодирования множества спектральных значений содержит инициализацию 310 контекста. Инициализация 310 контекста включает извлечение текущего контекста из предыдущего контекста с помощью функции "arith\_map\_context (N, arith\_reset\_flag)". Извлечение текущего контекста из предыдущего контекста может выборочно включать сброс контекста. Как сброс контекста, так и извлечение текущего контекста из предыдущего контекста будут рассмотрены ниже.

Декодирование множества спектральных значений также включает повторение декодирования спектральных значений 312 и обновление контекста 313, обновление контекста 313 выполняется функцией "arith\_update\_context(i,a,b)", которая описана ниже.

Декодирование спектральных значений 312 и обновление контекста 312 повторяется  $\lg/2$  раз, при этом  $\lg/2$  указывает число двоек спектральных значений для декодирования (например, для аудио фрейма), пока не обнаруживается так называемый символ "ARITH\_STOP". Кроме этого, декодирование набора  $\lg$  спектральных значений также включает декодирование знаков 314 и итоговый шаг 315.

Декодирование двойки спектральных значений 312 включает расчет значения контекста 312a, декодирование наиболее значимой битовой плоскости 312b, арифметическое обнаружение конечного символа 312c, добавление менее значимой битовой плоскости 312d и обновление массива 312e.

Вычисление значения состояния 312a включает вызов функции "arith\_get\_context(c,i,N)", как показано, например, на фиг. 5c или 5d. Соответственно, числовое значение текущего контекста (состояния) с предоставляется как возвратное значение вызова функции "arith\_get\_context(c,i,N)". Таким образом, числовое значение предыдущего контекста (также обозначенное "c"), которое является входной переменной функции "arith\_get\_context(c,i,N)" обновляется для того, чтобы получить, в качестве возвратного значения, числовое значение текущего контекста c.

Декодирование наиболее значимой битовой плоскости 312b включает повторное выполнение алгоритма декодирования 312ba и извлечение 312bb значений a, b из полученного значения m алгоритма 312ba, при этом переменная lev инициализируется до нуля. Алгоритм 312ba повторяется до тех пор, пока не достигнута команда "прерывание" (или условие). Алгоритм 312ba включает вычисление индекса состояния „rki" (который также служит в качестве индекса сводной таблицы частот) в зависимости от числового значения текущего контекста c, а также в зависимости от значения уровня "esc\_nb", с помощью функции "arith\_get\_pk()", которая обсуждается ниже (соответствующие варианты реализации изобретения показаны, например, на фиг. 5e и 5f). Алгоритм 312ba также включает выбор сводной таблицы частот в зависимости от индекса состояния rki, который обновляется вызовом функции "arith\_get\_pk()", при этом переменная "sum freq" может быть установлена на начальный адрес одной из 96 сводных таблиц частот в зависимости от индекса "rki". Переменная "cfl" может быть

инициализирована на длину выбранной сводной таблицы частот (или части таблицы), которая, например, равна количеству символов в алфавите, т.е. количеству различных значений, которые могут быть декодированы. Длины всех сводных таблиц частот (или частей таблиц) от "arith\_cf\_m[pki=0][17]" до "arith\_cf\_m[pki=95][17]", доступных для декодирования значения наиболее значимой битовой плоскости  $t$ , составляют 17, так что 16 различных значений наиболее значимых битовых плоскостей и управляющий символ ("ARITH\_ESCAPE") могут быть декодированы. Впоследствии, значение  $m$  наиболее значимой битовой плоскости может быть получено путем выполнения функции "arith\_decode()", с учетом выбранной сводной таблицы частоты (описанной переменной "cum+freq" и переменной "cf1"). При извлечении значения  $m$  наиболее значимой битовой плоскости, оцениваются биты под названием "acod\_m" в битовом потоке 210 (см., например, фиг. 6g или 6h).

Алгоритм 312ba также включает проверку того, равно ли значение наиболее значимой битовой плоскости  $m$  управляющему символу "ARITH\_ESCAPE", или нет. Если значение наиболее значимой битовой плоскости  $m$  не равно арифметическому управляющему символу, алгоритм 312ba прерывается (условие "перерывания"), а остальные инструкции алгоритма 312ba поэтому могут быть пропущены. Таким образом, выполнение процесса продолжается установкой значения  $b$  и значения  $a$  на шаг 312bb. В отличие от этого, если декодированное значение  $m$  наиболее значимой битовой плоскости совпадает с арифметическим управляющим символом "ARITH\_ESCAPE", значение уровня „lev" увеличивается на единицу. Значение уровня "esc\_nb" приравнивается к значению уровня „lev" пока переменная „lev" не превышает значение 7, в этом случае переменная "esc\_nb" устанавливается в значение 7. Как уже упоминалось, алгоритм 312ba повторяется до тех пор, пока декодированное значение  $m$  наиболее значимой битовой плоскости отличается от арифметического управляющего символа, при этом используется модифицированный контекст (поскольку входной параметр функции "arith\_get\_pk()") адаптируется в зависимости от значения переменной значения "esc\_nb").

Так как наиболее значимая битовая плоскость декодируется при помощи однократного или повторяющегося выполнения алгоритма 312ba, т.е. декодируется значение  $m$  наиболее значимой битовой плоскости, отличающееся от арифметического управляющего символа, переменная спектрального значения "b" приравнивается множеству (например, 2) наиболее значимых битов значения  $m$  наиболее значимой битовой плоскости, и переменная спектрального значения "a" устанавливается в соответствии с (например, 2) наименьшими битами значения  $m$  наиболее значимой битовой плоскости. Подробности относительно функциональности можно рассмотреть с помощью ссылки 312bb.

Следовательно, на шаге 312 с проверяется, присутствует ли арифметический символ остановки. Это тот случай, когда значение  $m$  наиболее значимой битовой плоскости равно нулю, а переменная „lev" больше, чем ноль. Соответствен, арифметическое условие остановки обозначается как «необычное» условие, при котором значение  $m$  наиболее значимой битовой плоскости равно нулю, а переменная „lev" указывает, что значению  $m$  наиболее значимой битовой плоскости соответствует увеличенный числовой вес. Другими словами, арифметическое условие остановки обнаруживается тогда, когда битовый поток указывает, что увеличенный числовой вес, больший, чем минимальный числовой вес, должен соответствовать значению наиболее значимой битовой плоскости, которое равно нулю, что является условием, которое не встречается в нормальной ситуации кодирования. Другими словами, обнаруживается арифметическое условие остановки, если кодированный арифметический управляющий символ перехода следует

за кодированным значением 0 наиболее значимой битовой плоскости.

После оценки присутствия арифметического условия остановки, которая выполняется на шаге 212с, могут быть получены менее значимые битовые плоскости, например, как показано на ссылке с номером 212d на фиг. 3. Для каждой менее значимой битовой плоскости декодируются два бинарных значения. Одно бинарное значение соответствует переменной *a* (или первому спектральному значению двойки спектральных значений) и другое бинарное значение соответствует переменной *b* (или второму спектральному значению двойки спектральных значений). Количество менее значимых битовых плоскостей обозначено переменной *lev*.

При декодировании одной или более менее значимых битовых плоскостей (если они присутствуют) повторно выполняется алгоритм 212da, при этом количество выполнений алгоритма 212da определяется переменной "lev". Необходимо отметить, что первый повтор алгоритма 212da выполняется на основе значений переменных *a*, *b*, как это настроено на шаге 212bb. Дальнейшие повторы алгоритма 212da выполняются на основе обновленных значений переменных *a*, *b*.

В начале повтора выбирается таблица сводных частот. Следовательно, выполняется арифметическое декодирование для того, чтобы получить значение переменной *r*, при этом значение переменной *r* описывает множество менее значимых битов, например, один менее значимый бит соответствует переменной *a*, другой менее значимый бит соответствует переменной *b*. Функция "ARITH\_DECODE" используется для того, чтобы получить значение *r*, при этом для арифметического декодирования используется сводная таблица частот "arith\_cf\_r".

Следовательно обновляются значения переменных *a* и *b*. Для этой цели переменная *a* сдвигается влево на один бит, и наименее значимый бит переменной *a*, которая была сдвинута, устанавливается как значение, определяемое как наименее значимый бит значения *r*. Переменная *b* сдвигается влево на один бит, и наименее значимый бит переменной *b*, которая была сдвинута, устанавливается как значение, определяемое как бит 1 переменной *r*, при этом бит 1 переменной *r* имеет числовой вес 2 в бинарном представлении переменной *r*. Алгоритм 412ba повторяется до тех пор, пока не будут декодированы все наименее значимые биты.

После декодирования менее значимых битовых плоскостей обновляется массив "x\_ac\_dec", в котором значения переменных сохраняются в записях указанного массива, имея индексы массива  $2*i$  и  $2*i+1$ .

Следовательно обновляется состояние контекста при использовании функции "arith\_update\_context(i,a,b)", которая подробнее будет рассмотрена далее со ссылкой на фиг. 5g.

За обновлением состояния контекста, который выполняется на шаге 313, повторяются алгоритмы 312 и 313 до тех пор, пока текущая переменная *i* не достигнет значения  $lg/2$  или не будет обнаружено арифметическое условие остановки.

Далее выполняется завершающий алгоритм "arith\_finish()", как показывает ссылка 315. Подробнее завершающий алгоритм "arith\_finish()" будет рассматриваться далее со ссылкой на фиг. 5m.

После завершающего алгоритма 315 декодируются знаки спектральных значений с помощью алгоритма 314. Как можно увидеть, знаки спектральных значений, отличных от нуля, кодируются индивидуально. С помощью алгоритма 314 читаются знаки всех спектральных значений, имеющие индексы *i* между  $i=0$  и  $i=lg-1$ , которые не равны нулю. Для каждого неравного нулю спектрального значения, имеющего индекс спектрального значения между  $i=0$  и  $i=lg-1$ , читается значение (обычно единичный бит) *s* из битового

потока. Если значение  $s$ , которое читается из битового потока, равно 1, знак указанного спектрального значения инвертируется. Для этой цели получается доступ к массиву "x\_ac\_dec" для того, чтобы определить, равно ли нулю спектральное значение, имеющее индекс  $i$ , а также для того, чтобы обновить знаки декодированных спектральных значений. Однако необходимо отметить, что знаки переменных  $a$ ,  $b$  остаются неизменными слева при декодировании знака 314.

При выполнении завершающего алгоритма 315 перед декодированием знаков 314 возможно сбросить все необходимые элементы после символа ARITH\_STOP.

Необходимо отметить, что концепция получения значений менее значимых битовых плоскостей не имеет особого значения для некоторых вариантов реализации настоящего изобретения. В некоторых вариантах реализации настоящего изобретения можно опустить декодирование менее значимых битовых плоскостей. В качестве альтернативы для этой цели могут быть использованы другие алгоритмы декодирования.

#### 11.2 Порядок декодирования в соответствии с фиг. 4

Далее будет описан порядок декодирования спектральных значений.

Квантованные спектральные коэффициенты "x\_ac\_dec[]" бесшумно кодируются и передаются (например, в битовом потоке), начиная с самого низкочастотного коэффициента и переходя к самому высокочастотному коэффициенту.

Следовательно, квантованные спектральные коэффициенты "x\_ac\_dec[]" бесшумно декодируются, начиная с самого низкочастотного коэффициента и переходя к самому высокочастотному коэффициенту. Квантованные спектральные коэффициенты декодируются группами из двух следующих друг за другом (например, смежных по частоте) коэффициентов  $a$  и  $b$ , которые собираются в иак называемые двойки ( $a$ ,  $b$ ) (также обозначенные как  $\{a, b\}$ ). Необходимо отметить, что квантованные спектральные коэффициенты иногда обозначаются как "qdec".

Декодированные коэффициенты "x\_ac\_dec[]" затем сохраняются для режима частотной области (например, декодированные коэффициенты для усовершенствованного аудио кодирования, например, полученные с помощью модифицированного дискретного косинус преобразования, как описано в ISO/IEC 14496, часть 3, раздел 4) в массиве "x\_ac\_quant[g][win][sfb][bin]". Порядок передачи кодовых слов бесшумного кодирования таков, что, когда они декодируются в порядке поступления и хранения в массиве, "bin" является наиболее быстро увеличивающимся индексом и "g" является наиболее медленно увеличивающимся индексом. В пределах кодового слова порядок декодирования будет  $a$ ,  $b$ .

Декодированные коэффициенты "x\_ac\_dec[]" хранятся для преобразования кодированного возбуждения (TCX), например, непосредственно в массиве  $x\_tcx\_invquant[win][bin]$ , а порядок передачи кодовых слов бесшумного кодирования такой, что, когда они декодируются в порядке поступления и хранения в массиве, "bin" является наиболее быстро увеличивающимся индексом и "win" является наиболее медленно увеличивающимся индексом. В пределах кодового слова порядок декодирования будет  $a$ ,  $b$ . Другими словами, если спектральные значения описывают преобразование кодированного возбуждения фильтра линейного предсказания кодера речи, спектральные значения  $a$ ,  $b$  связаны со смежными и увеличивающимися частотами преобразования кодированного возбуждения. Спектральные коэффициенты, соответствующие более низкой частоте, кодируются перед спектральными коэффициентами, соответствующими более высокой частоте.

Примечательно, что аудио декодер 200 может быть настроен на применение декодированного аудио представления в частотной области 232, которое обеспечивается

арифметическим декодером 230, как для "прямой" генерации представления аудио сигнала во временной области с помощью преобразования сигнала из частотной области во временную область, так и для "косвенного" предоставления представления аудио сигнала во временной области, используя как декодер из частотной области во временную область, так и фильтр линейного предсказания, возбуждаемый выходом трансформера сигнала из частотной области во временную область.

Другими словами, арифметический декодер, функции которого обсуждаются здесь подробно, хорошо подходит для декодирования спектральных значений представления аудио содержания во временной и частотной области, кодированного в частотной области, и для обеспечения представления стимула сигнала во временной и частотной области для фильтра линейного предсказания, адаптированного для декодирования (или синтеза) речевого сигнала, кодированного в области линейного предсказания. Таким образом, арифметический декодер хорошо подходит для использования в аудио декодере, способном работать как с аудио содержанием, кодированным в частотной области, так и с аудио содержанием, кодированным в линейно предсказанной частотной области (режим преобразования кодированного возбуждения области линейного предсказания).

### 11.3 Инициализация контекста в соответствии с фиг. 5a и 5b

Далее будет описана инициализация контекста (также обозначается как "отображение контекста"), которая выполняется в шаге 310.

Инициализация контекста включает сопоставление между прошлым контекстом и текущим контекстом в соответствии с алгоритмом "arith\_map\_context()", первый пример показан на фиг. 5a второй пример показан на фиг. 5a.

Как видно, текущий контекст хранится в глобальной переменной  $q[2][n\_context]$ , которая принимает форму массива, имеющего первое измерение 2 и второе измерение "n\_context". Простой контекст при необходимости (но это не обязательно) может храниться в переменной "qs[n\_context]", которая принимает форму таблицы, имеющей измерение "ncontext" (если она используется).

Согласно примеру алгоритма "arith\_map\_context" на фиг. 5a, входная переменная N описывает длину текущего окна, а входная переменная "arith\_reset\_flag" указывает, необходим ли сброс контекста. Кроме этого, глобальная переменная "previous\_N" описывает длину предыдущего окна. Необходимо отметить, что обычно число спектральных значений, соответствующих окну, по меньшей мере приблизительно равняется половине длины указанного окна в плане образцов временной области. Кроме этого, необходимо отметить, что число двоек спектральных значений, соответственно, по меньшей мере приблизительно равняется половине длины указанного окна в плане образцов временной области.

Согласно примеру на фиг. 5a, отображение контекста может выполняться в соответствии с алгоритмом "arith\_map\_context()". Необходимо отметить, что функция "arith\_map\_context()" устанавливает записи  $q[0][i]$  массива текущего контекста q в нулевое значение для  $i=0$  до  $i=N/4-1$ , если флаг "arith\_reset\_flag" является активным и указывает, что контекст должен быть сброшен. В противном случае, если флаг "arith\_reset\_flag" является неактивным, записи  $q[0][i]$  массива текущего контекста q извлекаются из записей  $q[1][k]$  массива текущего контекста q. Необходимо отметить, что функция "arith\_map\_context()" согласно фиг. 5a устанавливает записи  $q[0][i]$  массива текущего контекста q в значения  $q[1][k]$  массива текущего контекста q, если количество спектральных значений, соответствующих текущему (например, кодированному в частотной области) аудио фрейму, равняется количеству спектральных значений,



соответствующих предыдущему аудио фрейму для  $j=k=0$  до  $j=k=N/4-1$ .

Более сложное отображение выполняется, если количество спектральных значений, соответствующих текущему аудио фрейму, отличается от количества спектральных значений, соответствующих предыдущему аудио фрейму. Однако подробности, касающиеся отображения в данном случае, не особенно важны для ключевой идеи настоящего изобретения, так что за более подробной информацией можно обратиться к ссылке на псевдо программный код на фиг. 5a.

Кроме этого, инициализация значения для числового значения с текущего контекста возвращается с помощью функции "arith\_map\_context()". Это значение инициализации может быть равным, например, значению записи  $q[0][0]$ , сдвинутому влево на 12 битов. Таким образом, числовое значение  $s$  (текущего) контекста оказывается инициализированным для повторного обновления.

Кроме этого, на фиг. 5b показан другой пример алгоритма "arith\_map\_context()", который может быть использован альтернативным образом. За более подробной информацией можно обратиться к ссылке на псевдо программный код на фиг. 5b.

Таким образом, флаг "arith\_reset\_flag" определяет, должен ли быть сброшен контекст. Если флаг активен, вызывается субалгоритм сброса 500 а алгоритма "arith\_map\_context ()". Наоборот, если флаг "arith\_reset\_flag" неактивен (что означает отсутствие необходимости сброса контекста), процесс декодирования начинается со стадии инициализации, где вектор (или массив) элемента контекста обновляется при помощи копирования и отображения элементов контекста предыдущего фрейма, которые сохраняются в  $q[1][ ]$  до  $q[0][ ]$ . Элементы контекста в пределах  $q$  сохраняются на 4 битах на каждую двойку. Копирование и/или отображение элемента выполняется субалгоритмом 500b.

В примере, показанном на фиг. 5b, процесс декодирования начинается со стадии инициализации, где выполняется отображение сохраненным прошлым контекстом, который сохраняется в  $qs$ , и контекстом текущего фрейма  $q$ . Прошлый контекст  $qs$  сохраняется на 2 битах на каждую линию частоты.

#### 11.4 Вычисление значения состояния в соответствии с фиг. 5c и 5d

Далее будет более подробно описано вычисление значения состояния 312a.

Первый образец алгоритма будет описан со ссылкой на фиг. 5c, второй образец алгоритма будет описан со ссылкой на фиг. 5d.

Следует отметить, что числовое значение  $s$  с текущего контекста (как показано на фиг. 3) может быть получено в качестве возвращаемого значения функции "arith\_get\_context(c,i,N)", представление псевдо программного кода которого показано на фиг. 5c. В качестве альтернативы числовое значение  $s$  с текущего контекста может быть получено в качестве возвращаемого значения функции "arith\_get\_context(c,i)", представление псевдо программного кода которого показано на фиг. 5d.

Что касается вычисления значения состояния, также делается ссылка на фиг. 4, которая показывает контекст, используемый для оценки состояния, т.е. для вычисления числового значения  $s$  с текущего контекста. Фиг. 4 показывает двумерное представление спектральных значений как по времени, так и по частоте. Абсцисса 410 описывает время, а ордината 412 описывает частоту. Как видно на фиг. 4 кортеж спектральных значений 420, которые подвергаются декодированию (предпочтительно с использованием числового значения текущего контекста), связан с индексом времени  $t_0$  и индексом частоты  $i$ . Как видно, для индекса времени  $t_0$ , кортежи, имеющие индексы частоты  $i-1$ ,  $i-2$  и  $i-3$ , уже декодированы, в то время как спектральные значения кортежа 120 с индексом частоты  $i$  должны быть декодированы. Как видно из фиг. 4, спектральное

значение 430, имеющее индекс времени  $t_0$  и индекс частоты  $i-1$ , уже декодировано до того, как будет декодирован кортеж 420 спектральных значений, а кортеж 430 спектральных значений рассматривается для контекста, который используется для декодирования кортежа 420 спектральных значений. Таким же образом, кортеж 440 спектральных значений, имеющий индекс времени  $t_0-1$  и индекс частоты  $i-1$ , кортеж 450 спектральных значений, имеющий индекс времени  $t_0-1$  и индекс частоты  $i$ , кортеж 460 спектральных значений, имеющий индекс времени  $t_0-1$  и индекс частоты  $i+1$ , уже декодированы до того, как будет декодирован кортеж 420 спектральных значений, а также учтены для определения контекста, который используется для декодирования кортежа 420 спектральных значений. Спектральные значения (коэффициенты), уже декодированные в то время, когда декодируются и рассматриваются для контекста спектральные значения кортежа 420, показаны в заштрихованных квадратах. В отличие от этого, некоторые другие спектральные значения, уже декодированные (в то время, когда декодируются спектральные значения кортежа 420), но не рассмотренные для контекста (для декодирования спектральных значений кортежа 420), представлены квадратами с пунктирными линиями, а также другие спектральные значения (которые до сих пор не декодированы в то время, когда декодируются спектральные значения кортежа 420) показаны окружностями с пунктирными линиями. Кортежи, представленные квадратами с пунктирными линиями, и кортежи, представленные окружностями с пунктирными линиями, не используются для определения контекста для декодирования спектральных значений кортежа 420.

Тем не менее, следует отметить, что некоторые из этих спектральных значений, которые не используются для "обычного" или "нормального" вычисления контекста для декодирования спектральных значений кортежа 420, могут, тем не менее, быть оцененными для выявления множества ранее декодированных смежных спектральных значений, которые выполняют, по отдельности или вместе взятые, заданное условие относительно их величин. Подробнее этот вопрос будет рассматриваться далее.

Алгоритм "arith\_get\_context(c,i,N)" будет подробно рассмотрен со ссылкой на фиг. 5с. На фиг. 5с показано функционирование указанной функции "arith\_get\_context(c,i,N)" в виде псевдо программного кода, который использует условные обозначения известного языка C и/или C++. Таким образом, будут рассмотрены некоторые детали вычисления числового значения "c" текущего контекста, которое осуществляется с помощью функции "arith\_get\_context(c,i,N)".

Следует отметить, что функция "arith\_get\_context(c,i,N)" получает, в качестве входных переменных, «старый контекст состояния», который может быть описан с помощью числового значения предыдущего контекста "c". Функция "arith\_get\_context(c,i,N)" также получает, в качестве входной переменной, индекс  $i$  кортежа из двух декодируемых спектральных значений. Индекс  $i$ , как правило, является индексом частоты. Входная переменная  $N$  описывает длину окна, для которого декодируются спектральные значения.

Функция "arith\_get\_context(c,i,N)" обеспечивает, в качестве выходного значения, обновленную версию входной переменной  $c$ , которая описывает обновленный контекст состояния и которая может быть рассмотрена как числовое значение текущего контекста. Таким образом, функция "arith\_get\_context(c,i,N)" получает числовое значение "c" предыдущего контекста как входную переменную и предоставляет ее обновленную версию, которая рассматривается как числовое значение текущего контекста. В дополнение к этому функция "arith\_get\_context(c,i,N)" учитывает переменные  $i$ ,  $N$ , а также получает доступ к «глобальному» массиву  $q[][]$ .

Что касается функции "arith\_get\_context(c,i,N)", необходимо отметить, что переменная "с", которая изначально представляет числовое значение предыдущего контекста в бинарной форме, сдвигается вправо на 4 бита на шаге 504а. Соответственно, отбрасываются четыре наименее значимых бита числового значения предыдущего контекста (представленного с помощью входной переменной с). Также уменьшается числовой вес остальных битов числовых значений предыдущего контекста, например, с коэффициентом 16.

Кроме этого, если индекс  $i$  кортежа из 2-х элементов меньше чем  $N/4-1$ , т.е он не принимает максимальное значение, числовое значение текущего контекста модифицируется таким образом, что значение записи  $q[0][i+1]$  добавляется к битам с 12 по 15 (т.е. к битам, имеющим числовой вес  $2^{12}$ ,  $2^{13}$ ,  $2^{14}$  и  $2^{15}$ ) значения контекста, которое сдвигалось, что выполняется на шаге 504а. Для этой цели запись  $q[0][i+1]$  массива  $q[[]]$  (или, если быть более точным, бинарная репрезентация значения, представленного указанной записью) сдвигается влево на 12 битов. Измененная версия значения, представленная записью  $q[0][i+1]$  затем добавляется к значению контекста с, которое извлекается на шаге 504а, т.е. к числовому представлению со сдвинутыми битами (т.е. сдвинутому влево на 4 бита) числового значения предыдущего контекста. Необходимо отметить, что запись  $q[0][i+1]$  массива  $q[[]]$  представляет значение поддиапазона, соответствующее предыдущей части аудио содержания (например, часть аудио содержания, имеющая временной индекс  $t_0-1$ , что было определено при рассмотрении фиг. 4), а также имеющее более высокую частоту (например, частоту, имеющую индекс частоты  $i+1$ , что было определено при рассмотрении фиг. 4), чем кортеж спектральных значений, декодируемых в настоящий момент (используя числовое значение текущего контекста с на выходе функции "arith\_get\_context(c,i,N)"). Другими словами, если кортеж 420 декодируемых спектральных значений декодируется при помощи числового значения текущего контекста, запись  $q[0][i+1]$  может быть основана на кортеже 460 ранее декодированных спектральных значений.

Необходимое добавление записи  $q[0][i+1]$  массива  $q[[]]$  (сдвинутого влево на 12 битов) показано с помощью ссылки под номером 504b. Как можно видеть, добавление значения, представленного записью  $q[0][i+1]$  выполняется только в том случае, если индекс частоты  $i$  не обозначает кортеж спектральных значений, имеющий самый высокий индекс частоты  $i=N/4-1$ .

Следовательно на шаге 504с выполняется булева операция AND, при которой значение переменной с объединяется посредством операции AND с шестнадцатеричным значением 0xFFF0 для того, чтобы получить значение переменной с. При выполнении операции AND четыре наименее значимых бита переменной с устанавливаются в 0.

На шаге 504d значение записи  $q[0][i+1]$  добавляется к значению переменной с, которая получена на шаге 504с, для того, чтобы обновить значение переменной с. Однако, указанное обновление переменной с на шаге 504d выполняется только в том случае, если индекс частоты  $i$  декодируемого кортежа из 2-х значений больше нуля. Необходимо отметить, что запись  $q[1][i-1]$  является значением поддиапазона контекста, основанным на кортеже ранее декодированных спектральных значений текущей части аудио содержания для частот, меньших чем частоты спектральных значений, декодируемых при помощи числового значения текущего контекста. Например, запись  $q[1][i-1]$  массива  $q[[]]$  может соответствовать кортежу 430, имеющему временной индекс  $t_0$  и частотный индекс  $i-1$ , если предполагается, что кортеж 420 спектральных значений должен быть декодирован при помощи числового значения текущего контекста, возвращенного при выполнении функции "arith\_get\_context(c,i,N)".

Таким образом, биты 0, 1, 2 и 3 (т.е. четыре наименее значимых бита) числового значения предыдущего контекста отбрасываются на шаге 504a при помощи сдвига за пределы бинарного числового представления числового значения предыдущего контекста. Кроме этого, биты 12, 13, 14 и 15 сдвинутой переменной  $s$  (т.е. сдвинутого

5 числового значения предыдущего контекста) настроены принимать значения, определенные значением поддиапазона контекста  $q[0][i+1]$  на шаге 504b. Биты 0, 1, 2 и 3 сдвинутого числового значения предыдущего контекста (т.е. биты 4, 5, 6 и 7 исходного числового значения предыдущего контекста) переписываются значением поддиапазона контекста  $q[0][i+1]$  на шаге 504c и 504d.

10 Следовательно, можно сказать, что биты от 0 до 3 числового значения предыдущего контекста представляют значение поддиапазона контекста, соответствующее кортежу 432 спектральных значений, биты от 4 до 7 числового значения предыдущего контекста представляют значение поддиапазона контекста, соответствующее кортежу 434 ранее декодированных спектральных значений, биты от 8 до 11 числового значения

15 предыдущего контекста представляют значение поддиапазона контекста, соответствующее кортежу 440 ранее декодированных спектральных значений, и биты от 12 до 15 числового значения предыдущего контекста представляют значение поддиапазона контекста, соответствующее кортежу 450 ранее декодированных спектральных значений. Числовое значение предыдущего контекста на входе функции

20 "arith\_get\_context(c,i,N)" соответствует декодированию кортежа 430 спектральных значений.

Числовое значение текущего контекста, получаемое как переменная на выходе функции "arith\_get\_context(c,i,N)", соответствует декодированию кортежа 420 спектральных значений. Соответственно, биты с 0 по 3 числовых значений текущего

25 контекста описывают значение поддиапазона контекста, соответствующее кортежу 430 спектральных значений, биты с 4 по 7 числовых значений текущего контекста описывают значение поддиапазона контекста, соответствующее кортежу 440 спектральных значений, биты с 8 по 11 числовых значений текущего контекста описывают значение поддиапазона контекста, соответствующее кортежу 450

30 спектральных значений, и биты с 12 по 15 числовых значений текущего контекста описывают значение поддиапазона контекста, соответствующее кортежу 460 спектральных значений. Таким образом, можно увидеть, что часть числового значения предыдущего контекста, а именно, биты с 8 по 15 числового значения предыдущего контекста, также включены в числовое значение текущего контекста, как и биты с 4 по 11

35 числового значения текущего контекста. И наоборот, биты с 0 по 7 текущего числового значения предыдущего контекста отбрасываются при извлечении числового представления числового значения текущего контекста из числового представления предыдущего контекста.

На шаге 504e переменная  $s$ , которая представляет числовое значение текущего

40 контекста, при необходимости обновляется, если индекс частоты  $i$  декодируемого кортежа из 2-х значений больше, чем заданное значение, например, 3. В этом случае, т.е. если  $i$  больше 3, определяется, не превышает ли (или равняется) сумма значений поддиапазона контекста  $q[1][i-3]$ ,  $q[1][i-2]$  и  $q[1][i-1]$  заданное значение, например, 5. Если обнаружено, что сумма указанных значений поддиапазона контекста меньше,

45 чем указанное заданное значение, к переменной  $s$  добавляется шестнадцатеричное значение, например,  $0 \times 10000$ . Соответственно, переменная  $s$  устанавливается таким образом, что она указывает, есть ли условие, при котором значения поддиапазона контекста  $q[1][i-3]$ ,  $q[1][i-2]$  и  $q[1][i-1]$  составляют особенно малое значение суммы.

Например, бит 16 числового значения текущего контекста может выступать флагом для индикации такого состояния.

Итак, возвратное значение функции "arith\_get\_context(c,i,N)" определяется с помощью шагов 504a, 504b, 504c, 504d и 504e, где извлекается числовое значение текущего контекста из числового значения предыдущего контекста на шаге 504a, 504b, 504c, 504d и 504e, при этом на шаге 504e извлекается и добавляется к переменной с флаг, указывающий на окружение ранее декодированных спектральных значений, имеющих, в среднем, особенно малые абсолютные значения. Соответственно, значение переменной с, полученное на этапах 504a, 504b, 504c, 504d, возвращается на шаге 504f в качестве возвратного значения функции "arith\_get\_context(c,i,N)", если не выполняется условие, которое было оценено на шаге 504e. И наоборот, значение переменной с, полученное на этапах 504a, 504b, 504c, 504d, увеличивается на шестнадцатеричное значение  $0 \times 10000$ , и результат операции увеличения возвращается на шаге 504e, если выполняется условие, которое было оценено на шаге 504e.

Подводя итог вышесказанному, необходимо отметить, что бесшумный декодер имеет на выходе кортеж из двух беззнаковых квантованных спектральных коэффициентов (что будет подробно рассматриваться далее). Сначала состояние с контекста вычисляется на основе ранее декодированных спектральных коэффициентов, «окружающих» декодируемый кортеж из 2-х элементов. В предпочтительном варианте реализации изобретения состояние (которое представлено, например, с помощью числового значения контекста) обновляется с увеличением при помощи состояния контекста последнего декодированного кортежа из 2-х элементов (обозначенного как числовое значение предыдущего контекста), учитывая только два новых кортежа из 2-х элементов (например, кортежи 430 и 460). Состояние кодируется на 17 битах (например, используя числовое представление числового значения текущего контекста) и возвращается функцией "arith\_get\_context()". Более подробно это рассматривается на фиг. 5 с, где представлен код программы.

Необходимо отметить, что код псевдо программы альтернативного варианта функции "arith\_get\_context()" показан на фиг. 5d. Функция "arith\_get\_context()" согласно фиг. 5d подобна функции "arith\_get\_context(c,i,N)" согласно фиг. 5с. Однако функция "arith\_get\_context(c,i)" согласно фиг. 5d не включает обработку или декодирование кортежей спектральных значений, содержащих минимальный индекс частоты  $i=0$  или максимальный индекс частоты  $i=N/4-1$ .

#### 11.5 Выбор правила отображения

Далее будет описан выбор правила отображения, например, сводной таблицы частот, которая описывает отображение значения кода на код символа. Выбор правила отображения производится в зависимости от состояния контекста, который описывается числовым значением текущего контекста с.

##### 11.5.1 Выбор правила отображения с помощью алгоритма в соответствии с Фиг. 5е

Далее описывается выбор правила отображения с помощью функции "arith\_get\_pk(c)". Следует отметить, что функция "arith\_get\_pk()" вызывается в начале субалгоритма 312ba при декодировании значения кода "acod\_m" для получения кортежа спектральных значений. Необходимо отметить, что функция "arith\_get\_pk(c)" вызывается с различными аргументами при различных повторах алгоритма 312b. Например, при первом повторе алгоритма 312b функция "arith\_get\_pk(c)" вызывается с аргументом, равным числовому значению текущего контекста с, который получен при предыдущем выполнении функции "arith\_get\_context(c,i,N)" на шаге 312a. Наоборот, при дальнейших повторах субалгоритма 312a, вызывается функция "arith\_get\_pk(c)" с аргументом, который представляет собой

сумму числового значения текущего контекста  $s$ , полученного функцией "arith\_get\_context(c,i,N)" на шаге 312a, и версии значения переменной "esc\_nb" со сдвинутыми битами, при этом значение переменной "esc\_nb" сдвинуто влево на 17 битов. Таким образом, числовое значение текущего контекста  $s$ , полученное функцией "arith\_get\_context(c,i,N)"  
 5 используется в качестве входного значения функции "arith\_get\_pk()" при первом повторе алгоритма 312ba, т.е. при декодировании сравнительно небольших спектральных значений. Наоборот, при декодировании сравнительно больших спектральных значений входная переменная функции "arith\_get\_pk()" модифицируется с учетом значения переменной "esc\_nb", как показано на фиг. 3.

10 При рассмотрении фиг. 5e, на которой показан код псевдо программы первого варианта реализации функции "arith\_get\_pk(c)", необходимо отметить, что функция "arith\_get\_pk()" получает переменную  $s$  в качестве входного значения, при этом переменная  $s$  описывает состояние контекста, и при этом входная переменная  $s$  функции "arith\_get\_pk()" равняется числовому значению текущего контекста, полученному в  
 15 качестве возвратной переменной функцией "arith\_get\_context()" по меньшей мере в некоторых ситуациях. Кроме этого, необходимо отметить, что функция "arith\_get\_pk()" обеспечивает, в качестве выходной переменной, переменную "pk $i$ ", которая описывает индекс модели вероятности и может рассматриваться как значение индекса правила отображения.

20 При рассмотрении фиг. 5e можно видеть, что функция "arith\_get\_pk()" включает инициализацию переменной 506a, при этом инициализируется переменная "i\_min", принимая значение -1. Подобным образом переменная  $i$  приравнивается к переменной "i\_min", так что переменная  $i$  также инициализируется со значением -1. Переменная "i\_max" инициализируется и принимает значение, меньшее на 1, чем число записей  
 25 таблицы "ari\_lookup\_m[]" (которая будет более подробно рассмотрена со ссылкой на фиг. 21(1) и 22(2)). Соответственно, переменные "i\_min" и "i\_max" определяются как интервал.

Следовательно, поиск 506b выполняется для идентификации значения индекса, который обозначает запись таблицы "ari\_hash\_m", так что значение входной переменной  
 30  $s$  функции "arith\_get\_pk()" находится в пределах интервала, определенного указанной записью и смежной записью.

При поиске 506b субалгоритм 506ba повторяется, пока разница между переменными "i\_min" и "i\_max" больше, чем 1. В субалгоритме 506ba переменная  $i$  устанавливается  
 35 равной среднеарифметическому значений переменных "i\_min" и "i\_max". Следовательно переменная  $i$  обозначает запись таблицы "ari\_hash\_m[]" в середине интервала таблицы, который определен переменными "i\_min" и "i\_max". Следовательно, переменная  $j$  приравнивается значению записи "ari\_hash\_m[i]" таблицы "ari\_hash\_m()". Таким образом, переменная  $j$  принимает значение, определенное записью таблицы "ari\_hash\_m[]", которая находится в середине интервала таблицы, определенного с помощью переменных "i\_min"  
 40 и "i\_max". Следовательно, интервал, определенный переменными "i\_min" и "i\_max", обновляется, если значение входной переменной  $s$  функции "arith\_get\_pk()" отличается от значения состояния, определенного наивысшими битами записи таблицы "j=ari\_hash\_m[i]" таблицы "ari\_hash\_m[]". Например, «верхние биты» (8 и выше) записей таблицы "ari\_hash\_m[]" описывают значимые значения состояния. Соответственно, значение  
 45 "j>>8" описывает значимое значение состояния, представленное записью "j=ari\_hash\_m[i]" таблицы "ari\_hash\_m[]" и обозначенное записью индекса хэш-таблицы  $i$ . Соответственно, если значение переменной  $s$  меньше, чем значение "j>>8", это означает, что значение состояния, описанного переменной  $s$ , меньше, чем значение значимого

состояния, описанного с помощью записи "ari\_hash\_m[i]" таблицы "ari\_hash\_m[]". В этом случае значение переменной "i\_max" приравнивается значению переменной i, которая, в свою очередь, сокращает размер интервала, определенного с помощью переменных "i\_min" и "i\_max", при этом новый интервал приблизительно равняется нижней половине предыдущего интервала. Если обнаруживается, что входная переменная с функции "arith\_get\_pk()" больше, чем значение "j>>8", что означает, что значение контекста, описанного с помощью переменной c больше, чем значение значимого состояния, описанного с помощью записи "ari\_hash\_m[i]" массива "ari\_hash\_m[]", значение переменной "i\_min" приравнивается значению переменной i. Соответственно, размер интервала, определенного с помощью значений переменных "i\_min" и "i\_max", уменьшается до приблизительно половины размера предыдущего интервала, определенного с помощью предыдущих значений переменных "i\_min" и "i\_max". Если быть более точным, интервал, определенный обновленным значением переменной "i\_min" и предыдущим (неизменным) значением переменной "i\_max", приблизительно равен большей половине предыдущего интервала в том случае, если значение переменной c больше, чем значение значимого состояния, определенного с помощью записи "ari\_hash\_m[i]".

Однако, если обнаруживается, что значение контекста, описанного с помощью входной переменной с алгоритма "arith\_get\_pk()" равняется значимому значению состояния, определенному с помощью записи "ari\_hash\_m[i]" (т.е.  $c=(j \gg 8)$ ), значение индекса правила отображения, определенного с помощью самых нижних 8 битов записи "ari\_hash\_m[i]", возвращается в качестве возвратного значения функции "arith\_get\_pk()" (инструкция "return (j&0xFF)").

Суммируя вышесказанное, необходимо отметить, что запись "ari\_hash\_m[i]", наибольшие биты (биты от 8 и выше) которой описывают значение значимого состояния, оценивается при каждом повторе 506ba, и значение контекста (или числовое значение текущего контекста), описанное при помощи входной переменной с функции "arith\_get\_pk()", сравнивается со значением значимого состояния, описанного с помощью указанной записи таблицы "ari\_hash\_m[i]". Если значение контекста, представленное с помощью входной переменной c, меньше, чем значение значимого состояния, представленное записью таблицы "ari\_hash\_m[i]", верхняя граница (описанная с помощью значения "i\_max") интервала таблицы уменьшается, и если значение контекста, представленное с помощью входной переменной c, больше, чем значение значимого состояния, представленное записью таблицы "ari\_hash\_m[i]", нижняя граница (описанная с помощью значения "i\_max") интервала таблицы увеличивается. В обоих указанных случаях субалгоритм 506ba повторяется, пока размер интервала (определенного разностью между "i\_min" и "i\_max") не станет меньше или равным 1. Если, наоборот, значение контекста, описанного с помощью переменной c, равняется значению значимого состояния, описанного с помощью записи "ari\_hash\_m[i]", функция "arith\_get\_pk()" прерывается, при этом возвратное значение определяется с помощью наименьших битов записи таблицы "ari\_hash\_m[i]".

Однако, если поиск 506b прекращается из-за того, что размер интервала достигает своего минимального значения ("i\_min" - "i\_max" меньше или равно 1), возвратное значение функции "arith\_get\_pk()" определяется с помощью записи "ari\_lookup\_m[i\_max]" таблицы "ari\_lookup\_m[]", как можно увидеть с помощью ссылки 506c. Соответственно, записи таблицы "ari\_hash\_m[]" определяют как значимые значения состояния, так и границы интервалов. В субалгоритме 506b границы интервала поиска "i\_min" и "i\_max" повторно адаптируются таким образом, что запись "ari\_hash\_m[i]" таблицы "ari\_hash\_m

[], индекс хэш-таблицы  $i$  которой находится, по меньшей мере приблизительно, в центре интервала поиска, определенного с помощью значений границ интервала " $i_{\min}$ " и " $i_{\max}$ ", по меньшей мере приблизительно определяет значение контекста, описанного с помощью входной переменной  $s$ . Таким образом достигается, что значение контекста, описанное с помощью входной переменной  $s$ , находится в пределах интервала, определенного с помощью " $ari\_hash\_m[i_{\min}]$ " и " $ari\_hash\_m[i_{\max}]$ ", после завершения повторов субалгоритма 506ba, пока значение контекста, описанное с помощью входной переменной  $s$ , не будет равным значимому значению состояния, которое описано с помощью записи таблицы " $ari\_hash\_m[]$ ".

Однако, если повтор субалгоритма 506ba прекращается из-за того, что размер интервала (определенный при помощи " $i_{\min}$ " - " $i_{\max}$ ") достигает или превышает минимальное значение, считается, что значение контекста, описанное с помощью входной переменной  $s$ , не является значимым значением состояния. В этом случае все равно используется индекс " $i_{\max}$ ", который обозначает верхнюю границу интервала. Верхнее значение интервала " $i_{\max}$ ", которое получается при последнем повторе субалгоритма 506ba, используется повторно как значение индекса таблицы для доступа к таблице " $ari\_lookup\_m[]$ ". Таблица " $ari\_lookup\_m[]$ " описывает значения индекса правила отображения, соответствующие интервалам множества смежных числовых значений контекста. Интервалы, которым соответствуют значения индекса правила отображения, описанные с помощью записей таблицы " $ari\_lookup\_m[]$ ", определяются с помощью значимых значений состояния, которые описаны с помощью записей таблицы " $ari\_hash\_m[]$ ". Записи таблицы " $ari\_hash\_m[]$ " определяют как значимые значения состояния, так и границы интервалов смежных числовых значений контекста. При выполнении алгоритма 506b определяется, равно ли числовое значение контекста, описанное с помощью входной переменной  $s$ , значимому значению состояния, и, если оно не равно, то в каком интервале числовых значений контекста (из множества интервалов, границы которых определяются с помощью значимых значений состояния) находится значение контекста, описанное с помощью входной переменной  $s$ . Таким образом, алгоритм 506b выполняет двойную функцию, определяя, описывает ли входная переменная  $s$  значимое значение состояния, и, если нет, то идентифицируя интервал, ограниченный значимыми значениями состояния, в котором находится значение контекста, представленное входной переменной  $s$ . Соответственно, алгоритм 506b является особенно эффективным и требует сравнительно небольшое количество обращений к таблице.

Суммируя вышесказанное, необходимо отметить, что состояние контекста  $s$  определяет сводную таблицу частот, которая применялась для декодирования наиболее значимой 2-х-битовой плоскости  $m$ . Отображение от  $s$  до соответствующего индекса сводной таблицы частот " $pk_i$ " выполняется функцией " $arith\_get\_pk()$ ". Код псевдо программы указанной функции " $arith\_get\_pk()$ " рассматривался со ссылкой на фиг. 5e.

Таким образом, значение  $m$  декодируется, используя функцию " $arith\_decode$ " (которая рассматривается подробнее далее) совместно со сводной таблицей частот " $arith\_cf\_m[pk_i][i]$ ", где " $pk_i$ " соответствует индексу (также обозначенному как значение индекса правила отображения), возвращенному функцией " $arith\_get\_pk()$ ", которая рассматривается со ссылкой на фиг. 5e.

#### 11.5.2 Выбор правила отображения с помощью алгоритма в соответствии с Фиг. 5f

Далее будет рассмотрен другой вариант реализации алгоритма выбора правила отображения " $arith\_get\_pk()$ " со ссылкой на фиг. 5f, где показан код псевдо программы такого алгоритма, который может быть использован для декодирования кортежа



спектральных значений. Алгоритм в соответствии с фиг. 5f может быть рассмотрен как оптимизированная версия (например, версия с оптимизированной скоростью) алгоритма "get\_pk()" или алгоритма "arith\_get\_pk()".

Алгоритм "arith\_get\_pk()" в соответствии с фиг. 5f принимает, в качестве входной переменной, переменную *s*, которая описывает состояние контекста. Входная переменная *s* может, например, представлять числовое значение текущего контекста.

Алгоритм "arith\_get\_pk()" обеспечивает, в качестве выходной переменной, переменную "pki", которая описывает индекс вероятностной дистрибуции (или модели вероятности), соответствующий состоянию контекста, которое описано с помощью входной переменной *s*. Переменная "pki" может, например, быть значением индекса правила отображения.

Алгоритм в соответствии с фиг. 5f включает определение содержания массива "i\_diff[]". Как можно видеть, первая запись массива "i\_diff[]" (имеющая индекс массива 0) равна 299 и следующие записи массива (имеющие индексы массива от 1 до 8) принимают значения 149, 74, 37, 18, 94, 2 и 1. Соответственно, уменьшается размер шага для выбора значения индекса хеш-таблицы "i\_min" при каждом повторе, так как записи массива "i\_diff[]" определяют указанные размеры шага. Более подробно это будет рассмотрено далее.

Однако могут быть выбраны различные размеры шагов, например, разное содержание массива "i\_diff[]", при этом содержание массива "i\_diff[]" может быть адаптировано к размеру хеш-таблицы "ari\_hash\_m[i]".

Необходимо отметить, что переменная "i\_min" инициализируется и принимает значение 0 в самом начале алгоритма "arith\_get\_pk()".

На шаге инициализации 508a инициализируется переменная *s* в зависимости от входной переменной *s*, при этом числовое представление переменной *s* сдвигается влево на 8 битов для того, чтобы получить числовое представление переменной *s*.

Следовательно, осуществляется табличный поиск 508b для того, чтобы идентифицировать значение индекса хеш-таблицы "i\_min" записи хеш-таблицы "ari\_hash\_m[]" таким образом, что значение контекста, описанное с помощью значения контекста *s*, находится в пределах интервала, ограниченного значением контекста, представленным записью хеш-таблицы "ari\_hash\_m[i\_min]", и значением контекста, представленным другой записью хеш-таблицы "ari\_hash\_m", при этом запись "ari\_hash\_m" является смежной (в плане значения индекса хеш-таблицы) по отношению к записи "ari\_hash\_m[i\_min]". Таким образом, алгоритм 508b позволяет определить значение индекса хеш-таблицы "i\_min", обозначающее запись "j=ari\_hash\_m[i\_min]" хеш-таблицы "ari\_hash\_m[]" таким образом, что запись хеш-таблицы "ari\_hash\_m[i\_min]" по меньшей мере приблизительно определяет значение контекста, которое представлено с помощью входной переменной *s*.

Табличный поиск 508b включает повторяющееся выполнение субалгоритма 508ba, при этом субалгоритм 508ba выполняется заданное количество повторов, например, 9 раз. На первом шаге субалгоритма 508ba переменная *i* приравнивается сумме значений переменной "i\_min" и значению записи таблицы "i\_diff[k]". Необходимо отметить, что *i* представляет собой подвижную переменную, которая увеличивается, начиная от исходного значения *k*=0, с каждым повтором субалгоритма 508ba. Массив "i\_diff[]" определяет заданные значения увеличения, при этом значения увеличения уменьшаются при увеличении индекса таблицы *k*, т.е. с увеличением числа повторов.

На втором шаге субалгоритма 508ba значение записи таблицы "ari\_hash\_m[]" копируется в переменную *j*. Предпочтительно, если наибольшие биты записей таблицы

"ari\_hash\_m[]" описывают значения значимого состояния числового значения контекста, и наименьшие биты (от 0 до 7) записей таблицы "ari\_hash\_m[]" описывают значения индекса правила отображения, соответствующие необходимым значимым значениям состояния.

5 На третьем шаге субалгоритма 508ba значение переменной *s* сравнивается со значением переменной *j*, и переменная "i\_min" выборочно устанавливается как значение "i+1", если значение переменной *s* больше, чем значение переменной *j*. Следовательно, первый, второй и третий шаги субалгоритма 508ba повторяются заданное количество раз, например, 9 раз. Таким образом, при каждом выполнении субалгоритма 508ba значение переменной "i\_min" увеличивается на "i\_diff[]"+1, только если значение контекста, описанное с помощью действующего в настоящий момент индекса хеш-таблицы i\_min+i\_diff[], меньше, чем значение контекста, описанное с помощью входной переменной *s*. Соответственно, значение индекса хеш-таблицы "i\_min" (повторно) увеличивается при каждом выполнении субалгоритма 508ba, только если значение контекста, описанное с помощью входной переменной *s* и, соответственно, переменной *s*, больше, чем значение контекста, описанное с помощью записи "ari\_hash\_m[i=i\_min+diff[k]]".

Кроме этого, необходимо отметить, что только единичное сравнение, а именно, сравнение, больше ли значение переменной *s*, чем значение переменной *j*, осуществляется при каждом исполнении субалгоритма 508ba. Соответственно, субалгоритм 508ba особенно эффективен с точки зрения вычислений. Кроме этого, необходимо отметить, что возможны разные результаты в плане итогового значения переменной "i\_min". Например, возможно, что значение переменной "i\_min" после итогового исполнения субалгоритма 512ba такое, что значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min]" меньше, чем значение контекста, описанное с помощью входной переменной *s*, и значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min+1]" больше, чем значение контекста, описанное с помощью входной переменной *s*. Наоборот, может случиться, что после итогового исполнения субалгоритма 508ba значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min-1]" будет меньше, чем значение контекста, описанное с помощью входной переменной *s*, и значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min]" будет больше, чем значение контекста, описанное с помощью входной переменной *s*. В качестве альтернативы, может быть, что значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min]" будет равным значению контекста, описанному с помощью входной переменной *s*.

Для этой цели обеспечивается определенное возвратное значение 508c. Переменная *j* принимает значение записи хеш-таблицы "ari\_hash\_m[i\_min]". Следовательно, определяется, больше ли значение контекста, описанное с помощью входной переменной *s* (а также с помощью переменной *s*), чем значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min]" (первый случай определен условием "*s>j*"), или значение контекста, описанное с помощью входной переменной *s*, меньше, чем значение контекста, описанное с помощью записи таблицы "ari\_hash\_m[i\_min]" (второй случай определен условием "*c<s>>j*"), или контекста, описанное с помощью входной переменной *s*, равно значению контекста, описанному с помощью записи таблицы "ari\_hash\_m [i\_min]" (третий случай).

В первом случае, ("*s>j*"), запись "ari\_lookup\_m[i\_min+1]" таблицы "ari\_lookup\_m[]", обозначенная с помощью значения индекса таблицы "i\_min+1" возвращается в качестве выходного значения функции "arith\_get\_pk()". Во втором случае (*c<(s>>j)*) запись

"ari\_lookup\_m[i\_min]" таблицы "ari\_lookup\_m[]", обозначенная с помощью значения индекса таблицы "i\_min" возвращается в качестве возвратного значения функции "arith\_get\_pk(n). В третьем случае (т.е. если значение контекста, описанное с помощью входной переменной с равно значимому значению состояния, описанному с помощью записи таблицы "ari\_hash\_m[i\_min]"), значение индекса правила отображения, описанное с помощью наименьших 8 битов записи хеш-таблицы "ari\_hash\_m[i\_min]", возвращается в качестве возвратного значения функции "arith\_get\_pk()".

Суммируя вышесказанное, необходимо отметить, что на шаге 508b выполняется простой табличный поиск, при этом табличный поиск обеспечивает значение переменной "i\_min" без разграничения, равно или нет значению контекста, описанное с помощью входной переменной с, значимому значению состояния, определенному одной из записей состояния таблицы "ari\_hash\_m[i\_min]". На шаге 508 с, который выполняется вслед за табличным поиском 508b, оценивается отношение между величинами - значением контекста, которое описано с помощью входной переменной с, и значимым значением состояния, которое описано с помощью записи хеш-таблицы "ari\_hash\_m[i\_min]", а также выбирается возвратное значение функции "arith\_get\_pk()" в зависимости от результата указанной оценки, при этом значение переменной "i\_min", которое определяется при оценке таблицы 508b, учитывается при выборе значения индекса правила отображения, даже если значение контекста, описанное с помощью входной переменной с, отличается от значимого значения состояния, которое описано с помощью записи хеш-таблицы "ari\_hash\_m[i\_min]".

Необходимо дополнительно отметить, что сравнение в алгоритме должно выполняться предпочтительно (или в качестве альтернативы) между индексом контекста (числовое значение контекста) с  $j = \text{ari\_hash\_m}[i] \gg 8$ . В действительности каждая запись таблицы "ari\_hash\_m[]" представляет индекс контекста, кодированный за пределами 8 битов, а соответствующая ему вероятностная модель кодирована на первых 8 битах (наименее значимы биты). В данном варианте реализации изобретения важно выяснить, больше ли значение текущего контекста с, чем  $\text{ari\_hash\_m}[i] \gg 8$ , что эквивалентно выяснению, больше ли  $s = c \ll 8$ , чем  $\text{ari\_hash\_m}[i]$ .

Суммируя вышесказанное, необходимо отметить, что после вычисления состояния контекста (например, при помощи алгоритма "arith\_get\_context(c,i,N)" согласно фиг. 5c или при помощи алгоритма "arith\_get\_context(c,i)" согласно фиг. 5d) декодируется наиболее значимая 2-х-битовая плоскость при помощи алгоритма "arith\_decode" (который будет рассмотрен далее), который вызывается с подходящей сводной таблицей частот, соответствующей вероятностной модели, которая соответствует состоянию контекста. Соответствие согласовывается посредством функции "arith\_get\_pk()", например, функции "arith\_get\_pk()", которая рассматривалась со ссылкой на фиг. 5f.

## 11.6 Арифметическое декодирование

11.6.1 Арифметическое декодирование при помощи алгоритма в соответствии с фиг. 5g

Далее будет подробно рассмотрена функциональность функции "arith\_decode()" в соответствии с фиг. 5g.

Следует отметить, что функция "arith\_decode()" использует вспомогательную функцию "arith\_first\_symbol(void)", которая возвращает TRUE, если это первый символ последовательности и FALSE, если не первый. Функция "arith\_decode()" также использует вспомогательную функцию "arith\_get\_next\_bit(void)", которая получает и предоставляет следующий бит битового потока.

Кроме того, функция "arith\_decode()" использует глобальные переменные "low", "high"

и "value". Кроме того, функция "arith\_decode()" получает в качестве входной переменной, переменную "cum\_freq[]", которая указывает на первую запись или элемент (имеющий индекс элемента или индекс записи 0) выбранной сводной таблицы частот. Кроме того, функция "arith\_decode()" использует входную переменную "cfl", которая указывает на  
 5 длину выбранной сводной таблицы частот или дополнительную сводную таблицу частот, обозначенную переменной "cum\_freq[]".

Функция "arith\_decode()" включает в себя в качестве первого этапа инициализацию переменной 570a, которая выполняется, если вспомогательная функция "arith\_first\_symbol ()" показывает, что первый символ последовательности символов декодируется.

10 Инициализация значения 550a инициализирует переменную "value" в зависимости от множества, например, 16 бит, которые получаются из битового потока, используя вспомогательную функцию "arith\_get\_next\_bit", так, что переменная "value" имеет значение, представленное указанным числом бит. Кроме того, переменная "low" инициализируется, чтобы принять значение 0, а переменная "high" инициализируется,  
 15 чтобы принять значение 65535.

На втором этапе 570b переменная "range" устанавливается в значение, которое больше на 1, чем разница между значениями переменных "high" и "low". Переменная "cum" устанавливается в значение, которое представляет собой относительное положение значения переменной "value" между значением переменной "low" и значением переменной  
 20 "high". Таким образом, например, переменная "cum" принимает значение от 0 до  $2^{16}$  в зависимости от значения переменной "value".

Указатель *r* инициализируется в значение, которое меньше на 1, чем начальный адрес выбранной сводной таблицы частот.

25 Алгоритм "arith\_decode()" также включает итеративный поиск сводной таблицы частот 570c. Итеративный поиск сводной таблицы частот повторяется, пока переменная *cfl* меньше или равна 1. При итеративном поиске сводной таблицы частот 570c указатель переменной *q* устанавливается в значение, равное сумме текущего значения указателя переменной *r* и половине значения переменной "cfl". Если значение записи \**q* выбранной сводной таблицы частот, запись которой адресована указателем переменной *q*, больше,  
 30 чем значение переменной "cum", указатель переменной *r* устанавливается в значение указателя переменной *q*, и переменная "cfl" увеличивается. Наконец, переменная "cfl" смещается вправо на один бит, тем самым фактически разделяя значение переменной "cfl" на 2 и пренебрегая частью модуля.

35 Таким образом, итеративный поиск сводной таблицы частот 570 с фактически сравнивает значение переменной "cum" и множество записей выбранной сводной таблицы частот, чтобы определить интервал выбранной сводной таблицы частот, который ограничен записями сводной таблицы частот, так что значение *cum* находится в пределах выявленного интервала. Соответственно, записи выбранной сводной таблицы частот определяют интервалы, в которых соответствующие значения символа связаны  
 40 с каждым из интервалов выбранной сводной таблицы частот. Кроме того, размер ширины интервалов между двумя смежными значениями сводной таблицы частот определяет вероятности символов, соответствующих указанным интервалам, так что выбранная сводная таблица частот в целом определяет вероятность распределения разных символов (или значений символов). Подробнее доступные сводные таблицы  
 45 частот будут рассмотрены ниже, см. фиг. 23.

Возвращаясь к фиг. 5g, значение символа получено из значения переменной указателя *r*, при этом значение символа извлекается как показано на фиг. 570d. Таким образом, разница между значением переменной указателя *r* и начальным адресом "cum\_freq"

оценивается для того, чтобы получить значение символа, которое представлено переменной "symbol".

Алгоритм "arith\_decode" также включает в себя адаптацию 570e переменных "high" и "low". Если значение символа, представленное переменной "symbol", отличается от 0, переменная "high" обновляется, как показывает ссылка 570e. Кроме того, значение переменной "low" обновляется, как показано на ссылке с номером 570e. Переменная "high" устанавливается в значение, которое определяется значением переменной "low", переменной "range" и записью с индексом "symbol - 1" выбранной сводной таблицы частот. Переменная "low" увеличивается, причем величина роста определяется переменной "range" и записью выбранной сводной таблицы частот с индексом "symbol". Соответственно, разница между значениями переменных "low" и "high" регулируется в зависимости от числовой разницы между двумя смежными записями выбранной сводной таблицы частот.

Соответственно, если обнаружено значение символа, имеющее низкую вероятность, интервал между значениями переменных "low" и "high" сводится к малой ширине. Напротив, если обнаруженное значение символа содержит сравнительно большую вероятность, ширина интервала между значениями переменных "low" и "high" устанавливается в сравнительно большое значение. Ширина интервала между значениями переменных "low" и "high" зависит от обнаруженного символа и соответствующих записей сводной таблицы частот.

Алгоритм "arith\_decode()" также включает перенормировку интервала 570f, в котором интервал, определенный на шаге 570e, итеративно изменяется и масштабируется, пока не будет достигнуто условие "break". В перенормировке интервала 570f выполняется выборочная операция 570fa сдвига вниз. Если переменная "high" меньше, чем 32768, все остается как есть, а перенормировка интервала продолжает операцию увеличения размера интервала 570fb. Однако, если переменная "high" не меньше, чем 32768, а переменная "low" больше или равна 32768, переменные "values", "low" и "high" сокращаются на 32768, так что интервал, определенный переменными "low" и "high", смещается вниз, и так, что значение переменной "value" также сдвигается вниз. Однако, если будет установлено, что значение переменной "high" не меньше 32768, а также, что переменная "low" не превышает или равна 32768, а также, что переменная "low" больше или равна 16384 и, что переменная "high" меньше, чем 49152, переменные "value", "low" и "high" сокращаются на 16384, таким образом, смещая вниз интервал между значениями переменных "high" и "low", а также значение переменной "value". Если, однако, ни одно из указанных выше условий не выполняется, перенормировка интервала отменяется.

Однако, если любое из вышеуказанных условий, которые оцениваются на шаге 570fa, выполняется, выполняется операция увеличения интервала 570fb. В операции увеличения интервала 570fb значение переменной "low" удваивается. Кроме того, значение переменной "high" удваивается, и результат удвоения увеличивается на 1. Кроме того, удваивается значение переменной "value" (сдвигается влево на один бит), и бит битового потока, который получен вспомогательной функцией "arith\_get\_next\_bit", используется как наименее значимый бит. Соответственно, размер интервала между значениями переменных "low" и "high" приблизительно удваивается, и точность переменной "value" увеличивается за счет нового бита битового потока. Как уже упоминалось выше, шаги 570fa и 570fb повторяются, пока не выполнится условие "break", то есть, пока интервал между значениями переменных "low" и "high" достаточно велик.

Что касается функциональности алгоритма "arith\_decode()", следует отметить, что интервал между значениями переменных "low" и "high" сокращается на шаге 570e в

зависимости от двух смежных записей сводной таблицы частот, на которую ссылается переменная "cum\_freq". Если интервал между двумя смежными значениями выбранной сводной таблицы частот маленький, то есть, если смежные значения сравнительно близки друг к другу, интервал между значениями переменных "low" и "high", которые  
 5 получаются в шаге 570e, будет сравнительно небольшой. С другой стороны, если две смежные записи сводной таблицы частот расположены дальше, интервал между значениями переменных "low" и "high", который получается в шаге 570e, будет сравнительно большим.

Следовательно, если интервал между значениями переменных "low" и "high", который  
 10 получается в шаге 570e, сравнительно невелик, будет выполнено большое количество шагов перенормировки интервала, чтобы повторно отмасштабировать интервал к достаточному размеру (так, что ни одно из условий оценки 570fa не выполняется). Таким образом, сравнительно большое количество бит битового потока будет использовано для того, чтобы повысить точность переменной "value". Если, напротив,  
 15 размер интервала, полученного на шаге 570e, является сравнительно большим, потребуется меньшее количество повторений шагов перенормировки интервала 570fa и 570fb, чтобы перенормировать интервал между значениями переменных "low" и "high" до «достаточного» размера. Соответственно, будет использоваться лишь сравнительно небольшое количество бит битового потока, чтобы увеличить точность переменной  
 20 "value" и подготовить декодирование следующего символа.

Подводя итог вышесказанному, если символ декодирован, который содержит сравнительно высокую вероятность, и с которым связан большой интервал записей выбранной сводной таблицы частот, лишь сравнительно небольшое количество бит  
 25 будет считано из битового потока, с тем чтобы обеспечить декодирование последующих символов. С другой стороны, если символ декодирован, который содержит сравнительно небольшую вероятность, и с которым связан малый интервал записей выбранной сводной таблицы частот, из битового потока будет взято сравнительно большое количество бит, чтобы подготовить декодирование следующего символа.

Таким образом, записи сводных таблиц частот отражают вероятности разных  
 30 символов, а также отражают количество бит, необходимых для декодирования последовательности символов. Изменяя сводную таблицу частот в зависимости от контекста, т.е. в зависимости от ранее декодированных символов (или спектральных значений), например, путем выбора разных сводных таблиц частот в зависимости от  
 35 символами, что обеспечит особенно эффективное с точки зрения битрейта кодирование последующих (или смежных) символов.

Подводя итог вышесказанному, функция "arith\_decode()", которая была описана, ссылаясь на фиг. 5g, вызывается сводной таблицей частот "arith\_cf\_m[pki][ ]",  
 40 соответственно индексу "pki", возвращаемому функцией "arith\_get\_pk()", чтобы определить значение наиболее значимой битовой плоскости m (которое может быть установлено в значение символа, представляемого возвращаемой переменной "symbol").

Подводя итог вышесказанному, необходимо отметить, что арифметический декодер является целым вариантом реализации изобретения с использованием способа генерации тегов с масштабированием. Подробнее это рассматривается в работе "Introduction to  
 45 Data Compression" К. Sayood, Third Edition, 2006, Elsevier Inc.

Код компьютерной программы в соответствии с фиг. 5g описывает используемый алгоритм согласно варианту реализации настоящего изобретения.

11.6.2 Арифметическое декодирование с использованием алгоритма в соответствии

с фиг. 5h и 5i

На фиг. 5h и 5i показан код псевдопрограммы следующего варианта реализации алгоритма "arith\_decode()", который может быть использован как альтернатива алгоритму "arith\_decode", описанному со ссылкой на фиг. 5g.

5 Необходимо отметить, что оба алгоритма согласно фиг. 5g, 5h и 5i могут быть использованы в алгоритме "values\_decode()" согласно фиг. 3.

Таким образом, значение  $m$  декодируется с использованием функции "arith\_decode()", которая вызывается со сводной таблицей частот "arith\_cf\_m[pki][ ]", при этом "pki" соответствует индексу, возвращаемому функцией "arith\_get\_pk()". Арифметический  
10 кодер (или декодер) является целым вариантом реализации изобретения с использованием способа генерации тегов с масштабированием. Подробнее это рассматривается в работе "Introduction to Data Compression" К. Sayood, Third Edition, 2006, Elsevier Inc. Код компьютерной программы в соответствии с фиг. 5h и 5i описывает используемый алгоритм.

15 11.7 Механизм перехода

Далее будет кратко рассмотрен механизм перехода, который используется в алгоритме декодирования "values\_decode()" в соответствии с фиг. 3.

Когда декодированное значение  $m$  (которое возвращается как значение символа функцией "arith\_decode()") является символом перехода "ARITH\_ESCAPE", переменные  
20 "lev" и "esc\_nb" увеличиваются на 1 и декодируется другое значение  $m$ . В этом случае функция "arith\_get\_pk()" вызывается еще раз со значением "c+esc\_nb<<17" в качестве входного аргумента, где переменная "esc\_nb" описывает количество символов перехода, декодированных ранее для того же кортежа из 2-х элементов и ограниченного до 7.

Если идентифицируется символ перехода, то предполагается, что значение наиболее  
25 значимой битовой плоскости  $m$  содержит увеличенный числовой вес. Кроме этого повторяется текущее числовое декодирование, при этом модифицированное числовое значение текущего контекста "c+esc\_nb<<17" используется в качестве входной переменной функции "arith\_get\_pk()". Соответственно, обычно получаются различные значения индекса правила отображения "pki" при различных повторах субалгоритма  
30 312ba.

11.8 Арифметический механизм остановки

Далее будет рассмотрен арифметический механизм остановки. Арифметический механизм остановки позволяет сократить количество необходимых битов в том случае, если часть высоких частот полностью квантована аудио кодером до 0.

35 Арифметический механизм остановки может быть осуществлен следующим образом. Если значение  $m$  не является символом перехода "ARITH\_ESCAPE", декодер проверяет, не является ли последующее  $m$  символом "ARITH\_ESCAPE". Если выполняется условие "esc\_nb>0&& m==0", обнаруживается символ "ARITH\_STOP", и процесс декодирования останавливается. В этом случае декодер переходит непосредственно к функции  
40 "arith\_finish", которая будет рассмотрена далее. Условие означает, что остальная часть фрейма состоит из значений 0.

11.9 Декодирование менее значимой битовой плоскости

Далее будет рассмотрено декодирование одной или более менее значимой битовой плоскости. Декодирование менее значимой битовой плоскости осуществляется,  
45 например, на шаге 312d, как показано на фиг. 3. В качестве альтернативы могут быть использованы алгоритмы в соответствии с фиг. 5j и 5n.

11.9.1. Декодирование менее значимой битовой плоскости в соответствии с фиг. 5i

Как показано на фиг. 5j, значения переменных  $a$  и  $b$  извлекаются из значения  $m$ .

Например, числовое представление значения  $m$  сдвигается вправо на 2 бита для того, чтобы получить числовое представление переменной  $b$ . Кроме этого, значение переменной  $a$  получается посредством вычитания версии значения переменной  $B$ , смещенной влево на 2 бита, из значения переменной  $m$ .

5 Арифметическое декодирование значений  $g$  менее значимой битовой плоскости повторяются, при этом количество повторений определяется значением переменной "lev". Значение наименее значимой битовой плоскости  $g$  определяется при помощи функции "arith\_decode", при этом используется сводная таблица частот, адаптированная к декодированию наименее значимой битовой плоскости (сводная таблица частот  
10 "arith\_cf\_r"). Наименее значимый бит (имеющий числовой вес 1) переменной  $g$  описывает наименее значимую битовую плоскость  $g$  спектрального значения, представленного переменной  $a$ , и бит переменной  $g$ , имеющий числовой вес 2, описывает наименее значимый бит спектрального значения, представленного переменной  $b$ . Соответственно, переменная  $a$  обновляется посредством сдвига переменной  $a$  влево на 1 бит и  
15 прибавлением бита, имеющего числовой вес 1, переменной  $g$  в качестве наименее значимого бита. Подобным образом, переменная  $b$  обновляется посредством сдвига переменной  $b$  влево на 1 бит и прибавлением бита, имеющего числовой вес 2, переменной  $g$ .

Соответственно, два наиболее значимых бита переменных  $a$ ,  $b$ , несущие информацию,  
20 определяются с помощью значения наиболее значимой битовой плоскости  $m$ , и один или более наименее важный бит (если они присутствуют) значений  $a$  и  $b$  определяются одним или более значением  $g$  наименее значимой битовой плоскости.

Таким образом, если символ "ARITH\_STOP" не встречается, декодируются оставшиеся битовые плоскости, если они присутствуют, текущего кортежа из 2-х элементов.  
25 Оставшиеся битовые плоскости декодируются от уровня наиболее значимых к наименее значимым с помощью вызова функции "arith\_decode" в количестве lev раз совместно со сводной таблицей частот "arith\_cf\_r[]"). Декодированные битовые плоскости  $g$  допускают уточнение ранее декодированного значения  $m$  в зависимости от алгоритма, код псевдо программы которого показан на фиг. 5j.

30 11.9.2 Декодирование полосы наименее значимых битов в соответствии с фиг. 5n.

В качестве альтернативы алгоритм, код псевдо программы которого показан на фиг. 5n, может быть использован для декодирования наименее значимой битовой плоскости. В этом случае, если не встречается символ "ARITH\_STOP", декодируются оставшиеся битовые плоскости, если они присутствуют, текущего кортежа из 2-х  
35 элементов. Оставшиеся битовые плоскости декодируются от уровня наиболее значимых к наименее значимым с помощью вызова функции "arith\_decode" в количестве lev раз совместно со сводной таблицей частот "arith\_cf\_r()"). Декодированные битовые плоскости  $g$  допускают уточнение ранее декодированного значения  $m$  в зависимости от алгоритма, показанного на фиг. 5n.

40 11.10 Обновление контекста

11.11 Обновление контекста в соответствии с фиг. 5k, 5l и 5m.

Далее будут рассмотрены операции, используемые для завершения декодирования кортежа спектральных значений, со ссылкой на 5k и 5l. Кроме этого, будет рассмотрена операция, используемая для завершения декодирования набора кортежей спектральных  
45 значений, соответствующих текущей части (например, текущему фрейму) аудио содержания.

Как показано на фиг. 5k, запись, имеющая индекс  $2*i$  массива "x\_ac\_dec[]" приравнивается к  $a$ , и запись, имеющая индекс  $2*i+1$  массива "x\_ac\_dec[]" приравнивается



к b после декодирования менее значимого бита 312d. Другими словами, в момент после декодирования менее значимого бита 312d, беззнаковое значение кортежа из двух элементов (a, b) полностью декодируется. Оно сохраняется в элементе (например, в массиве "x\_ac\_dec[]"), который содержит спектральные коэффициенты в соответствии с алгоритмом, показанным на фиг. 5k.

Следовательно, контекст "q" также обновляется для следующего кортежа из 2-х элементов. Необходимо отметить, что обновление контекста также должно быть выполнено для последнего кортежа. Обновление контекста выполняется при помощи функции "arith\_update\_context()", код псевдо программы которой представлен на фиг. 5l.

Как показано на фиг. 5l, функция "arith\_update\_context(i,a,b)" получает в качестве входных переменных декодированные беззнаковые квантованные спектральные коэффициенты (или спектральные значения) a, b кортежа из 2-х элементов. Дополнительно функция "arith\_update\_context()" также принимает в качестве входной переменной для декодирования индекс i (например, частотный индекс) квантованного спектрального коэффициента. Другими словами, входная переменная i может, например, быть индексом кортежа спектральных значений, абсолютные значения которых определяются входными переменными a, b. Как можно увидеть, запись "q[1][i]" массива "q[][]" может быть настроена как значение, равное a+b+1. Дополнительно значение записи "q[1][i]" массива "q[][]" может быть ограничено до шестнадцатеричное значение "0xF". Таким образом, запись "q[1][i]" массива "q[][]" получается при вычислении суммы абсолютных значений текущего декодированного кортежа {a, b} спектральных значений, имеющих индекс частоты 1, и добавлении 1 к указанной сумме.

Необходимо отметить, что запись "q[1][i]" массива "q[][]" может рассматриваться как значение поддиапазона контекста, т.к. она описывает поддиапазон контекста, который используется для последующего декодирования дополнительных спектральных значений (или кортежей спектральных значений).

Необходимо отметить, что суммирование абсолютных значений a и b двух спектральных значений, декодированных в настоящий момент (знаковые версии сохраняются в записях "x\_ac\_dec[2\*i]" и "x\_ac\_dec[2\*i+1]"), может рассматриваться как вычисление нормы (например, нормы L1) декодированных спектральных значений.

Было обнаружено, что значения поддиапазона контекста (т.е. записи массива "q[][]"), которые описывают норму вектора, формируемого множеством ранее декодированных спектральных значений, являются особенно значимыми и эффективными с точки зрения занимаемой памяти. Было обнаружено, что норма, которая вычисляется на основе множества ранее декодированных спектральных значений, включает информацию о значимом контексте в компактной форме. Было обнаружено, что знак спектральных значений не особенно важен для выбора контекста. Также было обнаружено, что при формировании нормы на основе множества ранее декодированных спектральных значений обычно сохраняется самая важная информация, даже если опускаются некоторые детали. Кроме этого, было обнаружено, что сокращение числового значения текущего контекста до максимального значения обычно не приводит к значительной потере информации. Наоборот, было обнаружено, что более эффективно использовать то же самое состояние контекста для значимых спектральных значений, которые больше, чем заданное пороговое значение. Таким образом, сокращение значений поддиапазона контекста приводит к повышению эффективности с точки зрения занимаемой памяти. Кроме этого, было обнаружено, что сокращение значений поддиапазона контекста до определенного максимального

значения позволяет просто и эффективно с точки зрения вычислений обновлять числовое значение текущего контекста, что рассматривалось, например, со ссылкой на фиг. 5c и 5d. При сокращении значений поддиапазона контекста до сравнительно небольшого значения (например, до значения 15) состояние контекста, которое основано на

5 множестве значений поддиапазона контекста, может быть представлено эффективно, что рассматривалось со ссылкой на фиг. 5c и 5d.

Кроме этого, было обнаружено, что сокращение значений поддиапазона контекста до значений между 1 и 15 приводит компромиссу между точностью и затратами на память, т.к. для хранения подобного значения поддиапазона контекста достаточно 4-х

10 битов.

Однако, необходимо отметить, что в некоторых вариантах реализации изобретения значение поддиапазона контекста может быть основано только на одном декодированном спектральном значении. В этом случае формирование нормы может быть опущено.

15 Следующий кортеж из 2-х элементов фрейма декодируется после завершения функции "arith\_update\_context()" при помощи увеличения  $i$  на 1 и повторения процесса, как это было рассмотрено ранее, начиная с функции "arith\_get\_context()".

Когда декодируются  $lg/2$  кортежи из 2-х элементов в пределах фрейма или встречается символ остановки согласно "ARITH\_ESCAPE", процесс декодирования спектральной

20 величины завершается и начинается декодирование знаков.

Декодирование знаков подробно рассматривалось со ссылкой на фиг. 3, где декодирование знаков показано номером 314.

После декодирования всех беззнаковых квантованных спектральных коэффициентов добавляется знак взаимодействия. Для каждого квантованного значения "x\_ac\_dec",

25 неравного нулю, считывается бит. Если значение считанного бита равно 0, квантованное значение положительно, все остается как есть, и знаковое значение приравнивается к ранее декодированному беззнаковому значению. В противном случае (т.е. если значение считанного бита равно 1) декодированный коэффициент (или спектральное значение) является отрицательным и из беззнакового значения забирается дополнительный

30 двоичный код. Знаковые биты считываются от низких к высоким частотам. Более подробно это рассматривается со ссылкой на фиг. 3 и при объяснении относительно декодирования знаков 314.

Декодирование заканчивается вызовом функции "arith\_finish()". Оставшиеся спектральные коэффициенты устанавливаются в 0. Соответствующие состояния

35 контекста обновляются.

Более подробно это рассматривается со ссылкой на фиг. 5m, где показан код псевдо программы функции "arith\_finish()". Как можно увидеть, функция "arith\_finish()" принимает входную переменную  $lg$ , которая описывает декодированные квантованные спектральные коэффициенты. Входная переменная  $lg$  функции "arith\_finish()"

40 предпочтительно описывает ряд декодированных спектральных коэффициентов, оставляя неучтенными спектральные коэффициенты, которым присвоено значение 0 в ответ на обнаружение символа "ARITH\_STOP". Входная переменная  $N$  функции "arith\_finish()" описывает длину текущего окна (т.е. окна, соответствующего обрабатываемой части аудио содержания). Обычно число спектральных значений, соответствующих длине окна  $N$  равно  $N/2$ , а количество кортежей из 2-х спектральных значений, соответствующих окну с длиной  $N$ , равно  $N/4$ .

Функция "arith\_finish()" также получает в качестве входного значения вектор "xacdec" декодированных спектральных значений или по меньшей мере ссылку на вектор

декодированных спектральных значений.

Функция "arith\_finish()" настроена устанавливать записи массива (или вектор) "x\_ac\_dec", для которых не декодировались спектральные значения из-за присутствия арифметического условия остановки, в нулевое значение. Кроме этого, функция "arith\_finish()" настроена приравнивать значения поддиапазона контекста "q[1][i]", которые соответствуют спектральным значениям, для которых не декодировались значения из-за присутствия арифметического условия остановки, к заданному значению 1. Заданное значение 1 соответствует кортежу спектральных значений, в котором оба спектральных значения равны 0.

Соответственно, функция "arith\_finish()" позволяет обновлять весь массив (или вектор) "x\_ac\_dec" спектральных значений, а также весь массив значений поддиапазона контекста "q[1][i]", даже если присутствует арифметическое условие остановки.

#### 11.10.2 Обновление контекста в соответствии с фиг. 5о и 5р

Далее будет рассмотрен следующий вариант обновления контекста со ссылкой на фиг. 5о и 5р. После того, как беззнаковое значение кортежа из 2-х значений (a, b) полностью декодировано, контекст q обновляется для следующего кортежа. Обновление также выполняется в том случае, если текущий кортеж из 2-х значений является последним. Оба обновления выполняются с помощью функции "arith\_update\_context()", код псевдо программы которой показан на фиг. 5о.

Следующий кортеж из 2-х значений фрейма декодируется при увеличении i на 1 с помощью вызова функции "arith\_decode()". Если lg/2 кортежи из 2-х значений уже были декодированы с фреймом или встретился символ остановки "ARITH\_STOP", вызывается функция "arith\_finish()". Контекст сохраняется в массиве (или векторе) "qs" для следующего фрейма. Код псевдо программы функции "arith\_save\_context()" показан на фиг. 5р.

После того как декодированы все беззнаковые квантованные спектральные коэффициенты, добавляется знак. Для каждого неквантованного значения "qdec" считывается бит. Если значение считанного бита равно 0, квантованное значение является положительным, все остается как есть, и знаковое значение приравнивается к ранее декодированному беззнаковому значению. В противном случае декодированный коэффициент является отрицательным и из беззнакового значения забирается дополнительный двоичный код. Знаковые биты считываются от низких к высоким частотам.

#### 11.11 Обобщение процесса декодирования

Далее кратко описывается процесс декодирования. За более подробной информацией можно обратиться к вышеизложенному описанию, а также к фиг. 3, 4, 5а, 5с, 5е, 5g, 5j, 5k, 5l и 5m. Квантованные спектральные коэффициенты "x\_ac\_dec" бесшумно кодируются, начиная с самого низкого частотного коэффициента и увеличиваясь до самого высокого частотного коэффициента. Они декодируются в группах по два последующих коэффициента a, b, которые собираются в так называемые кортежи (a, b).

Декодированные коэффициенты "x\_ac\_dec" для частотной области (т.е. для режима частотной области) хранятся в массиве "x\_ac\_quant[g][win][sfb][bin]". Порядок передачи кодовых слов бесшумного кодирования таков, что, когда они декодируются в порядке получения и хранения в массиве, "bin" является наиболее быстро увеличивающимся индексом, а "g" самым медленно увеличивающимся индексом. В пределах кодового слова декодирование осуществляется в порядке a, затем b. Декодированные коэффициенты "x\_ac\_dec" для "ТСХ" (т.е. для аудио декодирования с использованием

преобразования кодированного возбуждения) сохраняются (например, напрямую) в массиве "x\_ac\_invquant[win][bin]", и порядок передачи кодовых слов бесшумного кодирования таков, что, когда они декодируются в порядке получения и хранения в массиве, "bin" является наиболее быстро увеличивающимся индексом, а "win" самым медленно увеличивающимся индексом. В пределах кодового слова декодирование осуществляется в порядке a, затем b.

Сначала флаг "arith\_reset\_flag" определяет, должен ли быть сброшен контекст. Если флаг является активным, это учитывается функцией "arith\_map\_context".

Процесс декодирования начинается со стадии инициализации, где обновляется элемент контекста вектор "q" с помощью копирования и отображения элементов контекста предыдущего фрейма, который сохранен в "q[1][]" до "q[0][]". Элементы контекста в пределах "q" сохраняются на 4-х битах на каждый кортеж из 2-х значений. За более подробной информацией можно обратиться к фиг. 5a, где показан код псевдо программы.

Бесшумный декодер предоставляет на выходе кортежи из 2-х беззнаковых квантованных спектральных коэффициентов. Сначала состояние контекста с вычисляется на основе ранее декодированных спектральных коэффициентов, окружающих декодируемый кортеж из 2-х элементов. Затем состояние обновляется с увеличением при помощи состояния контекста последних декодированных кортежей, учитывая только два новых кортежа. Состояние декодируется на 17 битах и возвращается функцией "arith\_get\_context". Код псевдо программы установленной функции "arith\_get\_context" показан на фиг. 5c.

Состояние контекста с определяет сводную таблицу частот, используемую для декодирования наиболее значимой 2-х битовой плоскости m. Отображение от с до соответствующего индекса сводной таблицы частот "pki" осуществляется функцией "arith\_get\_pki()". Код псевдо программы функции "arith\_get\_pki()" показан на фиг. 5e.

Значение m декодируется при помощи функции "arith\_decode()", вызываемой со сводной таблицей частот "arith\_cf\_m[pki][]", где "pki" соответствует индексу, возвращаемому функцией "arith\_get\_pki()". Арифметический кодер (и декодер) является целым вариантом реализации изобретения с использованием способа генерации тегов с масштабированием. Код псевдо программы, изображенный на фиг. 5g, описывает используемый алгоритм.

Когда декодированное значение m является символом перехода, "ARITH\_ESCAPE", переменные „lev" и "esc\_nb" увеличиваются на 1, другое значение m декодируется. В этом случае снова вызывается функция "get\_pk", имеющая значение "с+esc\_nb<<17" в качестве входного аргумента, где "esc\_nb" представляет количество, ограниченное до 7, символов перехода, декодируемых ранее для того же кортежа из 2-х элементов.

Если значение m не является символом перехода "ARITH\_ESCAPE", декодер проверяет, формирует ли последующее m символ "ARITH\_STOP". Если выполняется условие "(esc\_nb>0&&m==0)", обнаруживается символ "ARITH\_STOP" и процесс декодирования заканчивается. Декодер переходит непосредственно к знаковому декодированию, которое будет рассмотрено далее. Условие обозначает, что оставшая часть фрейма состоит из значений 0.

Если символ "ARITH\_STOP" не встречается, декодируются оставшиеся битовые плоскости, если таковые существуют, для данного кортежа из 2-х элементов. Оставшиеся битовые плоскости декодируются от наиболее значимого к наименее значимому уровню при помощи вызова "arith\_decode()" lev количество раз совместно со сводной таблицей частот "arith\_cf\_r[)". Декодированные битовые плоскости r позволяют уточнять ранее

декодированное значение  $m$  в соответствии с алгоритмом кода псевдо программы, который показан на фиг. 5j. На этом этапе беззнаковое значение кортежа  $(a, b)$  является полностью декодированным. Оно сохраняется в элементе, который содержит спектральные коэффициенты в соответствии с алгоритмом, код псевдо программы которой показан на фиг. 5k.

Контекст "q" также обновляется для следующего кортежа. Необходимо отметить, что обновление контекста также должно осуществляться для последнего кортежа. Обновление контекста выполняется функцией "arith\_update\_context()", код псевдо программы которой показан на фиг. 5l.

Следующий кортеж из 2-х элементов фрейма декодируется при помощи увеличения  $i$  на 1 и повтора описанного ранее процесса, начиная с функции "arith\_get\_context()". После того как  $lg/2$  кортежи декодированы в пределах фрейма или появляется символ остановки "ARITH\_STOP", процесс декодирования спектральной величины прекращается и начинается декодирование знаков.

Декодирование завершается вызовом функции "arith\_finish()". Оставшиеся спектральные коэффициенты приравниваются к 0. Соответствующие состояния контекста обновляются. Код псевдо программы функции "arith\_finish()" показан на фиг. 5m.

После декодирования всех беззнаковых квантованных спектральных коэффициентов, добавляется знак взаимодействия. Для каждого квантованного значения "x\_ac\_dec()", неравного нулю, считывается бит. Если значение считанного бита равно 0, то квантованное значение является положительным, все остается как есть, и знаковое значение приравнивается к ранее декодированному беззнаковому значению. В противном случае декодированный коэффициент является отрицательным, и из беззнакового значения забирается дополнительный двоичный код. Знаковые биты считываются от низких к высоким частотам.

#### 11.12 Условные обозначения

На фиг. 5q показана легенда определений, которые относятся к алгоритмам в соответствии с фиг. 5a, 5c, 5e, 5f, 5g, 5j, 5k, 5l и 5m.

На фиг. 5r показана легенда определений, которые относятся к алгоритмам в соответствии с фиг. 5b, 5d, 5f, 5h, 5i, 5n, 5o, и 5p.

#### 12 Таблицы отображения

В одном из вариантов осуществления изобретения особенно эффективные таблицы "ari\_lookup\_m", "ari\_hash\_m" и "ari\_cf\_m" используются для выполнения функции "arith\_get\_pk()" в соответствии с фиг. 5e или 5f, и для выполнения функции "arith\_decode", которая описывалась со ссылкой на фиг. 5g, 5h и 5i. Однако необходимо отметить, что в некоторых вариантах реализации изобретения могут быть использованы разные таблицы.

##### 12.1 Таблица "ari\_hash m [600]" в соответствии с фиг. 22

Содержание особо эффективного применения таблицы "ari\_hash\_m", которая используется функцией "arith\_get\_pk", первый вариант реализации которой показан со ссылкой на фиг. 5e и второй вариант реализации которой показан со ссылкой на фиг. 5f, представлено в таблице на фиг. 22. Следует отметить, что таблица на фиг. 22 содержит 600 записей таблицы (или массива) "ari\_hash m [600]". Следует также отметить, что таблица на фиг. 22 показывает элементы в порядке индексов элементов так, что первое значение "0x000000100UL" соответствует записи таблицы "ari\_hash\_m[0]", имеющей индекс элемента (или табличный индекс) 0, так, что последнее значение "0x7fffffff4fUL" соответствует записи таблицы "ari\_hash m [599]", имеющей индекс элемента или

табличный индекс 599. Далее следует отметить, что "0x" означает, что записи таблицы в таблице "ari\_hash m[]" представлены в шестнадцатеричном формате. Кроме того, необходимо отметить, что индекс "UL" указывает, что записи таблицы "ari\_hash m[]" представлены беззнаковыми «длинными» целыми значениями (имеющими точность 32 бита).

Далее необходимо отметить, что записи таблицы в таблице "ari\_hash m[]" на фиг. 22 расположены по числовому порядку, чтобы обеспечить выполнение табличного поиска 506b, 508b, 510b функции "arith\_get\_pk()".

Следует также отметить, что наиболее значимые 24 бита записи таблицы в таблице "ari\_hash m[]" представляют определенные значимые значения состояния, в то время как наименее значимые 8 бит представляют значения индекса правила отображения pki. Таким образом, записи таблицы "ari\_hash m[]" описывают отображение "прямого попадания" значения состояния в значении индекса правила отображения "pki".

Однако, наибольшие 24 бита записей таблицы "ari\_hash m[]" представляют, в то же самое время, границы интервалов числовых значений контекста, которым соответствует одно и то же значение индекса правила отображения. Данная концепция уже была подробно рассмотрена.

### 12.2 Таблица "ari\_lookup\_m" в соответствии с фиг. 21

Содержание особо эффективного варианта таблицы "ari\_lookup\_m" показано в таблице на фиг. 21. Следует отметить, что таблица на фиг. 21 содержит записи таблицы "ari\_lookup\_m". На указанные записи ссылается индекс одномерной записи целочисленного типа (также именуемая "индекс элемента" или "индекс массива" или "табличный индекс"), который, например, обозначается "i\_max" или "i\_min". Следует отметить, что таблица "ari\_lookup\_m", которая включает в себя всего 600 записей, хорошо подходит для использования функцией "arith\_get\_pk" в соответствии с фиг. 5e или 5f. Необходимо также отметить, что таблица "ari\_lookup\_m" в соответствии с фиг. 21 адаптирована для совместного функционирования с таблицей "ari\_hash m" в соответствии с фиг. 22.

Следует отметить, что записи таблицы "ari\_lookup\_m[600]" перечислены в порядке возрастания индекса табличного индекса i (например, "i\_max" или "i\_min") от нуля до 599. Термин "0x" означает, что записи в таблице приведены в шестнадцатеричном формате. Соответственно, первая запись таблицы "0x02" соответствует записи таблицы "ari\_lookup\_m[0]" с табличным индексом 0, и последняя запись таблицы "0x5E" соответствует записи таблицы "ari\_lookup\_m[599]" с табличным индексом 599.

Следует также отметить, что записи таблицы "ari\_lookup\_m[600]" соответствуют интервалам, которые определены смежными записями таблицы "arith\_hash m[]". Таким образом, записи таблицы "ari\_lookup\_m" описывают значения индекса правила отображения, соответствующие интервалам числовых значений контекста, при этом интервалы определяются при помощи записей таблицы "arith\_hash m[]".

### 12.3. Таблица "ari\_cf\_ml"961[171" в соответствии с фиг. 23

Фиг. 23 показывает набор из 96 сводных таблиц частот (или дополнительных таблиц) "ari\_cf\_m[pki][17]", одна из которых выбрана аудио кодером 100, 700, или аудио декодером 200, 800, например, для выполнения функции "arith\_decode()", то есть для декодирования значения наиболее значимой битовой плоскости. Выбранная одна из 96 сводных таблиц частот (или дополнительных таблиц), показанная на фиг. 23, выбирает функцию таблицы "cum\_freq []" для выполнения функции "arith\_decode()".

Как видно из фиг. 23, каждая подгруппа представляет собой сводную таблицу частот с 17 записями. Например, первая подгруппа 2310 представляет 17 записей сводной

таблицы частот для "pki=0". Вторая подгруппа 2312 представляет 17 записей сводной таблицы частот для "pki=1". Наконец, 96-я подгруппа 2396 представляет 17 записей сводной таблицы частот для "pki=95". Таким образом, фиг. 23 фактически представляет 96 различных сводных таблиц частот (или дополнительных таблиц) для "pki=0" до "pki=9", где каждая из 96 сводных таблиц частот представлена одной подгруппой (которая заключена в круглые скобки), и где каждая из указанных сводных таблиц частот включает 17 записей.

В подгруппе (например, подгруппе 2310, или подгруппе 2312, или подгруппе 2396) первое значение описывает первую запись сводной таблицы частот (имеющую индекс массива или индекс таблицы 0), и последнее значение описывает последнюю запись сводной таблицы частот (имеющую индекс массива или индекс таблицы 16).

Таким образом, каждая подгруппа 2310, 2312, 2396 таблицы на фиг. 23 представляет записи сводной таблицы частот для использования функцией "arith\_decode" в соответствии с фиг. 5g или в соответствии с фиг. 5h и 5i. Входная переменная "cum\_freq[]" функции "arith\_decode" описывает, какая из 96 сводных таблиц частот (представленных отдельными подгруппами из 17 записей таблицы "ari\_cf\_m") должна быть использована для декодирования текущих спектральных коэффициентов.

12.4 Таблица "ari\_cf\_r[]" в соответствии с фиг. 24

Фиг. 24 показывает содержание таблицы "ari\_cf\_r[]".

На фиг. 24 показаны четыре записи указанной таблицы. Однако, необходимо отметить, что таблицы "ari\_cf\_r[]" в разных вариантах реализации изобретения могут отличаться друг от друга.

13. Оценка функционирования и преимущества

Варианты реализации изобретения использует обновленные функции (или алгоритмы) и обновленный набор таблиц, как отмечалось выше, чтобы получить оптимальное соотношение между сложностью вычислений, требованиями к памяти, а также эффективностью кодирования.

В целом, варианты реализации настоящего изобретения производят улучшенное спектральное бесшумное кодирование. Варианты реализации настоящего изобретения представляют усовершенствование спектрального бесшумного кодирования в стандарте USAC (унифицированном кодировании речи и аудио).

Варианты реализации настоящего изобретения представляют обновленное предложение для SE по улучшению спектрального бесшумного кодирования спектральных коэффициентов, которое основано на схемах, представленных в документах MPEG ml6912 и ml7002. Оба предложения были проанализированы, потенциальные недостатки удалены, а сильные стороны объединены.

Как и в документах ml6912 и ml7002, итоговое предложение основано на "оригинальной" схеме арифметического кодирования на основе контекста, как описано в рабочем проекте 5 стандарта USAC (проект стандарта унифицированного кодирования речи и аудио), но существенно снижает требования к памяти (оперативной (RAM) и постоянной (ROM)) без увеличения сложности вычислений, осуществляя при этом эффективное кодирование. Кроме этого, была доказана возможность перекодировки битовых потоков без потерь в соответствии с рабочими проектами 3 и 5 проекта стандарта US AC.

Настоящее описание характеризует вариант осуществления для SE на улучшения спектрального бесшумного кодирования спектральных коэффициентов. Предложенная схема основана на "оригинальной" схеме арифметического кодирования на основе контекста, как описано в рабочем проекте 4 проекта стандарта USAC, но существенно

снижает требования к памяти (RAM, ROM), в то же время сохраняя бесшумное кодирование. Варианты реализации настоящего изобретения имеют своей целью заменить схему спектрального бесшумного кодирования как она используется в рабочем проекте 5 проекта стандарта USAC.

5       Описанная схема арифметического кодирования основана на схеме как в эталонной модели 0 (RM0) или рабочем проекте 4 (WD4) проекта стандарта USAC. Спектральные коэффициенты, расположенные по частоте или по времени, представляют собой модель контекста. Этот контекст используется для выбора сводных таблиц частот для арифметического кодера. По сравнению с рабочим проектом WD4 контекстное  
10       моделирование еще более усовершенствовано, и таблицы, содержащие вероятности символа, были повторно настроены. Число различных вероятностных моделей увеличилось с 32 до 96.

Варианты реализации изобретения уменьшают размеры таблицы (запрос на данные ROM) до 1518 слов длиной 32 бита или 6072 байт (WD: 16894,5 слов или 67578 байт).  
15       Статический запрос RAM снижается с 666 слов (2664 байт) до 72 (288 байт) на каждый основной канал кодера. В то же время, он полностью сохраняет выполнение кодирования и может даже достичь прироста приблизительно с 1,29% до 1,9%, по сравнению с общей скоростью передачи данных по всем 9 рабочим точкам. Все битовые потоки рабочих проектов 3 и 5 могут быть перекодированы без потерь, не влияя на  
20       ограничения битового резервуара.

Далее будет приведено краткое описание концепции кодирования в соответствии с рабочим проектом 5 проекта стандарта USAC для облегчения понимания преимуществ концепции, описанной в данном документе. А также будут рассмотрены предпочтительные варианты реализации настоящего изобретения.

25       В рабочем проекте 5 USAC схема контекстно зависимого арифметического кодирования используется для бесшумного кодирования квантованных спектральных коэффициентов. В качестве контекста используются декодированные спектральные коэффициенты, которые расположены ранее по частоте и времени. В соответствии с рабочим проектом 5 в качестве контекста используется максимальное количество из  
30       16-ти спектральных коэффициентов, 12 из которых расположены ранее по времени. Спектральные коэффициенты, используемые для контекста и для декодирования, сгруппированы в кортежи по 4 коэффициента (т.е. четыре спектральных коэффициента, соседних по частоте, см. фиг. 14а). Контекст сокращается и отображается в сводной таблице частот, которая затем используется для декодирования следующего кортежа  
35       из 4х спектральных коэффициентов.

Для полной схемы бесшумного кодирования согласно рабочему проекту 5 требуется запрос памяти (ROM) из 16894,5 слов (67578 байт). Кроме того, требуются 666 слов (2664 байт) статической оперативной памяти (RAM) на каждый основной канал кодера для хранения состояний для следующего фрейма. Таблица на фиг. 14b описывает  
40       таблицы, используемые в схеме арифметического кодирования USAC WD4.

Необходимо отметить, что в отношении бесшумного кодирования, рабочие проекты 4 и 5 проекта стандарта USAC одинаковы. Оба используют тот же самый бесшумный кодер.

Общий запрос памяти полного декодера USAC WD4 оценивается в 37000 слов (148000  
45       байт) для данных ROM без программного кода, и от 10000 до 17000 слов для статической оперативной памяти (RAM). Очевидно, что таблицы бесшумного кодера потребляют около 45% общего запроса данных постоянно памяти (ROM). Самая крупная отдельная таблица уже потребляет 4096 слов (16384 байт).



Было установлено, что и размер сочетания всех таблиц и отдельные крупные таблицы превышают типичные размеры кэша, содержащегося в процессорах с фиксированной точкой для портативных устройств потребительского класса, диапазон которых обычно составляет 8-32 кбайт (например, ARM9E, T1C64xx и т.д.). Это означает, что набор таблиц, вероятно, не может сохраняться в оперативной памяти (RAM), которая позволяет быстрый произвольный доступ к данным. Это приводит к тому, что весь процесс декодирования замедляется.

Кроме этого, было доказано, что современная эффективная технология аудио кодирования, такая как HE-AAC может применяться в большинстве мобильных устройств. HE-AAC применяет схему энтропийного кодирования Хаффмана с размером таблицы 995 слов. Более подробно см. ISO/IEC JTC1/SC29/WG11 N2005, MPEG98, February 1998, San Jose, "Revised Report on Complexity of MPEG-2 AAC2".

На 90-й конференции по проблемам MPEG в рабочих документах m16912 и m17002 были представлены два предложения, которые ставили своей целью сократить объем необходимой памяти и повысить эффективность схемы бесшумного кодирования. После анализа предложений модно прийти к следующим выводам.

Значительное уменьшение затрат на память возможно при сокращении размера кодового слова. Как показано в рабочем документе MPEG m17002, при сокращении размера кортежа с 4-х элементов до 1 элемента, затраты на память могут уменьшиться с 16984,5 до 900 слов без снижения эффективности кодирования; и

Может быть устранена избыточность при помощи применения для LSB кодирования кодовой книги неунифицированного вероятностного распределения вместо унифицированного вероятностного распределения.

В ходе проведенного анализа было обнаружено, что переход от схемы кодирования кортежей из 4-х элементов к кортежам из 1 элемента значительно влияет на уровень сложности вычислений: уменьшение размера кодируемого элемента увеличивает с тем же коэффициентом количество кодируемых символов. Это означает, что при уменьшении кортежа с 4-х элементов до 1 операции, необходимые для определения контекста, доступа к хэш-таблицам и декодирования символа, будут выполняться в четыре раза чаще, чем ранее. Кроме усложнения алгоритма по определению контекста это приводит к увеличению сложности вычислений на коэффициент 2,5 или  $x.xxPCU$ .

Далее будет кратко описана новая предлагаемая схема согласно вариантам реализации настоящего изобретения.

Для преодоления проблем, связанных с объемом необходимой памяти и сложностью вычислений, предлагается заменить схему рабочего проекта 5 (WD5) на улучшенную схему бесшумного кодирования. Основное внимание в данной разработке уделялось сокращению объема необходимой памяти, при сохранении достаточного сжатия данных и без повышения сложности вычислений. А именно, целью было достичь хорошего (или даже максимального) компромисса между осуществлением сжатия в многомерном сложном пространстве, сложностью вычислений и затратами на память.

Предложенная новая схема кодирования заимствует основной элемент бесшумного кодера WD5, а именно адаптацию контекста. Контекст извлекается при помощи ранее кодированных спектральных элементов, которые, как в WD5, поступают из прошлого и текущего фрейма (при этом фрейм может рассматриваться как часть аудио содержания). Спектральные коэффициенты кодируются при помощи комбинирования двух коэффициентов вместе в кортежи-двойки. Другое отличие состоит в том, что спектральные коэффициенты разделены на три части: знак, более значимые биты или наиболее значимые биты (MSBs) и менее значимые биты или наименее значимые биты

LSBs). Знак кодируется независимо от величины, которая далее разделяется на две части: наиболее значимые биты (или более значимые биты) и остальная часть битов (или менее значимые биты), если таковые присутствуют. Кортжи, у которых величина двух элементов меньше или равна 3, кодируются непосредственно с помощью MSBs кодирования. В противном случае сначала передается кодовое слово перехода, которое обозначает дополнительную битовую плоскость. В базовой версии отсутствующая информация, LSBs или знак, кодируются при помощи унифицированного распределения вероятностей. В качестве альтернативы могут использоваться различные распределения вероятностей.

10 Уменьшение размера таблицы возможно, т.к.:

необходимо сохранять только вероятности для 17 символов: {[0;+3], [0;+3]}+ESC символ;

отсутствует необходимость сохранять группирующую таблицу (egroups, dgroups, dgvector);

15 размер хэш-таблицы может быть уменьшен при определенной настройке.

Далее будут рассмотрены некоторые подробности относительно MSBs кодирования.

Как уже отмечалось, одним из основных различий между WD5 проекта стандарта USAC, предложением, представленным на 90-й Конференции по проблемам MPEG, и данным предложением является размер символов. В WD5 проекта стандарта USAC

20 рассматривались кортжи из 4-х элементов для создания контекста и бесшумного кодирования. В предложении, представленном на 90-й Конференции по проблемам MPEG, использовались кортжи из 1 элемента для уменьшения затрат ROM. В процессе разработки было обнаружено, что кортжи из 2-х элементов являются наилучшим решением для уменьшения затрат постоянной памяти (ROM) без увеличения сложности

25 вычислений. Вместо обработки четырех кортежей из 4-х элементов для создания контекста, теперь учитываются четыре кортежа из 2-х элементов. Как показано на фиг. 15а, три кортежа из 2-х элементов поступают из прошлого фрейма (также обозначенного как предыдущая часть аудио содержания) и один поступает из текущего фрейма (также обозначенного как текущая часть аудио содержания).

30 Уменьшение размера таблицы происходит благодаря трем основным факторам. Во-первых, необходимо сохранять только вероятности для 17 символов: {[0;+3], [0;+3]}+ESC символ). Отсутствует необходимость в группирующих таблицах (egroups, dgroups, dgvector). Наконец, размер хэш-таблицы уменьшается при определенной настройке.

Несмотря на то, что размер был уменьшен с четырех до двух, сохранился такой же

35 уровень сложности, как в WD5 проекта стандарта USAC. Это достигнуто путем упрощения как процесса создания контекста, так и доступа к хэш-таблице.

Упрощение и оптимизация выполнены так, что процесс кодирования не был затронут, а даже несколько улучшен. Это достигнуто, в основном, при помощи увеличения количества моделей вероятности с 32 до 96.

40 Далее будут рассмотрены некоторые подробности относительно LSBs кодирования. В некоторых вариантах реализации изобретения LSBs кодируются с помощью унифицированного распределения вероятности. В отличие от WD5 проекта стандарта USAC, LSBs обрабатываются кортежами из 2-х элементов вместо 4-х элементов.

Далее будут рассмотрены некоторые подробности относительно кодирования знака.

45 Знак кодируется без применения арифметического основного кодера с целью снижения сложности. Знак передается только одним битом, когда соответствующая величина не равна нулю. 0 обозначает положительное значение, а 1 обозначает отрицательное значение.

Далее будут рассмотрены вопросы, касающиеся требований к памяти. Предлагаемая новая схема показывает общий запрос ROM максимально 1522,5 новых слов (6090 байт). См. таблицу на фиг. 15b, которая описывает таблицы, используемые в предлагаемой схеме кодирования. В отличие от запроса ROM схемы бесшумного кодирования в WD5 проекта стандарта USAC, запрос ROM сокращается по меньшей мере на 15462 слова (61848 байт). Получается тот же порядок величин, который предъявляется для запроса памяти AAC декодера Хаффмана в HE-AAC (995 слов или 3980 байт). Более подробно см. ISO/IEC JTC1/SC29/WG11 N2005, MPEG98, February 1998, San Jose, "Revised Report on Complexity of MPEG-2 AAC2", а также фиг. 16a.

Это сокращает общий запрос ROM бесшумного кодера более чем на 92% и полного декодера USAC от примерно 37000 слов до примерно 21500 слов, или более чем на 41%. Более подробно см. фиг. 16a и 16b, при этом на фиг. 16a показаны требования к ROM в предложенной схеме бесшумного кодирования и в схеме бесшумного кодирования в соответствии с WD4 проекта стандарта USAC, а на фиг. 16b показан общий запрос на данные ROM декодера USAC в соответствии с предложенной схемой и в соответствии с WD4 проекта стандарта USAC.

Далее, количество информации, необходимой для вывода контекста в следующем фрейме (статическая RAM), также уменьшается. В соответствии с WD4 проекта стандарта USAC, необходимо сохранять полный набор коэффициентов (максимально 1152) с обычным разрешением 16 бит дополнительно к индексу группы на каждый кортеж из 4-х элементов разрешением 10 бит, что насчитывает 666 слов (2664 байт) на каждый основной канал кодера (полный декодер USAC WD4: приблизительно от 10000 до 17000 слов). Новая схема сокращает постоянную информацию до 2 бит на спектральный коэффициент, который насчитывает до 72 слов (288 байт) в сумме на основной канал кодера. Запрос на статическую память может быть сокращен на 594 слова (2376 байт).

Далее описываются некоторые подробности, касающиеся возможного повышения эффективности кодирования. Эффективность декодирования в соответствии с новым предложением вариантов реализации изобретения сравнивалась в качестве эталона с битовыми потоками в соответствии с рабочими проектами WD3 и WD5 проекта стандарта USAC. Сравнение проводилось с помощью транскодера на основе программного декодера в качестве эталона. Для дополнительной информации относительно сравнения бесшумного кодирования в соответствии с WD3 и WD5 проекта стандарта USAC и предлагаемой схемы кодирования см. фиг. 9, где схематично представлено тестирование.

Кроме этого, требования к памяти в вариантах реализации изобретения сравнивались с вариантами в соответствии с WD3 (или WD5) проекта стандарта USAC.

Эффективность кодирования не только сохраняется, но и несколько увеличивается. Более подробно см. таблицу на фиг. 18, где представлена таблица средних битрейтов, производимых арифметическим кодером WD3 (или аудио кодером USAC, который использует арифметический кодер WD3) и аудио кодером (например, аудио кодером USAC) в соответствии с вариантом реализации изобретения.

Информацию о средних битрейтах каждого оперативного режима можно найти в таблице на фиг. 18.

Кроме этого, на фиг. 19 представлена таблица минимального и максимального уровней резервуара бит для арифметического кодера WD3 (или аудио кодера USAC, который использует арифметический кодер WD3) и аудио кодера в соответствии с вариантом реализации настоящего изобретения.

Далее будут рассмотрены некоторые детали относительно сложности вычислений.

Уменьшение степени размерности арифметического кодирования обычно приводит к увеличению сложности вычислений. На самом деле, уменьшение размера вдвое приводит к увеличению в два раза операций арифметического кодера.

Однако, было обнаружено, что увеличение сложности может быть ограничено некоторыми модификациями, которые вводятся в новую схему кодирования в соответствии с реализацией настоящего изобретения. В некоторых вариантах реализации изобретения было значительно упрощено создание контекста. Для каждого кортежа из 2-х элементов контекст обновляется с увеличением на основе последнего созданного контекста. Вероятности сохраняются на 14 битах вместо 16, что позволяет избежать 64-битовые операции во время процесса декодирования. Кроме этого в некоторых вариантах реализации изобретения было значительно оптимизировано отображение вероятностной модели. Самый неблагоприятный случай был значительно улучшен и ограничен до 10 повторов вместо 95.

В результате сложность вычислений предложенной схемы бесшумного кодирования осталась на том же уровне, что и в WD5. Оценка «на бумаге» выполнялась различными версиями бесшумного кодирования и регистрировалась в таблице, которая представлена на фиг. 20. Она показывает, что новая схема кодирования только примерно на 13% менее сложна, чем арифметический кодер WD5.

В итоге можно увидеть, что варианты реализации настоящего изобретения обеспечивают оптимальный баланс между вычислительной сложностью, требованиями к памяти и эффективностью кодирования.

#### 14. Синтаксис битового потока

##### 14.1 Полезная нагрузка спектрального бесшумного кодера

Далее описываются некоторые детали, касающиеся полезной нагрузки спектрального бесшумного кодера. В некоторых вариантах есть множество различных режимов кодирования, таких как, например, так называемый режим кодирования «линейного предсказания области» и режим кодирования «частотной области». В режиме кодирования линейного предсказания области ограничение шума производится на основе анализа линейного предсказания аудио сигнала, и шумоподобный сигнал кодируется в частотной области. В режиме кодирования частотной области ограничение шума осуществляется на основе психоакустического анализа и шумоподобная версия аудио содержания кодируется в частотной области.

Спектральные коэффициенты кодированного сигнала «линейного предсказания области» и кодированного сигнала "частотной области" скалярно квантуются, а затем бесшумно кодируются при помощи адаптивного контекстно-зависимого арифметического кодирования. Квантованные коэффициенты собираются в кортежи-двойки до того, как они передаются от самых низких частот к самым высоким частотам. Каждый кортеж-двойка разделяется на знак  $s$ , наиболее значимую 2-х-битовую плоскость  $m$  и оставшиеся одну или более менее значимые битовые плоскости  $g$  (если таковые присутствуют). Значение  $m$  кодируется в соответствии со смежными коэффициентами. Другими словами,  $m$  кодируется в соответствии с окружением. Остальные менее значимые битовые плоскости  $g$  энтропийно-кодируются без учета контекста. Посредством  $m$  и  $g$  величина этих спектральных коэффициентов может быть реконструирована на стороне декодера. Для всех символов, неравных 0, знаки  $s$  кодируются за пределами арифметического кодера с использованием 1 бита. Другими словами, значения  $m$  и  $g$  формируют символы арифметического кодера. Таким образом, знаки  $s$  кодируются за пределами арифметического кодера с использованием 1 бита на каждый квантованный коэффициент, неравный 0.

Подробная процедура арифметического кодирования описана в данном документе.

#### 14.2. Элементы синтаксиса

Далее описывается синтаксис битового потока битового потока, несущего арифметически кодированную спектральную информацию, со ссылкой на фиг. 6а - 6j.

5 Фиг. 6а показывает синтаксическое представление так называемого блока необработанных данных USAC ("usac\_raw\_data\_block").

Блок необработанных данных USAC состоит из одного или более одноканальных элементов ("single\_channel\_element()") и/или одного или более двухканальных элементов ("channel\_pair\_element()").

10 Что касается фиг. 6b, то здесь описывается синтаксис одноканального элемента. Одноканальный элемент состоит из потока канала линейного предсказания области ("lpd\_channel\_stream()") или потока канала частотной области ("fd\_channel\_stream()") в зависимости от основного режима.

Фиг. 6с представляет синтаксис двухканального элемента. Двухканальный элемент 15 включает информацию об основном режиме ("core\_mode0", "core\_model"). Кроме того, двухканальный элемент может включать информацию о конфигурации "ics\_info()". Кроме того, в зависимости от информации об основном режиме двухканальный элемент состоит из потока канала линейного предсказания области или потока канала частотной области, связанного с первым из каналов, и двухканальный элемент также включает 20 поток канала линейного предсказания области или поток канала частотной области, связанный со вторым из каналов.

Информация о конфигурации "ics\_info()", синтаксическое представление которой показано на фиг. 6d, содержит множество различных элементов информации о конфигурации, которые нерелевантны для настоящего изобретения.

25 Поток канала частотной области ("fd\_channel\_stream()"), синтаксическое представление которого показано на фиг. 6е, включает получение информации ("global\_gain") и информации о конфигурации ("ics\_info()"). Кроме того, поток канала частотной области содержит данные коэффициента масштабирования ("scale\_factor\_data()"), которые описывают коэффициенты масштабирования, используемые для масштабирования 30 спектральных значений разных полос коэффициентов масштабирования, и который применяется, например, блоком масштабирования 150 и ресейлером 240. Поток канала частотной области также включает арифметически кодированные спектральные данные ("ac\_spectral\_data()"), которые представляют арифметически кодированные спектральные значения.

35 Арифметически кодированные спектральные данные ("ac\_spectral\_data()"), синтаксическое представление которых показано на фиг. 6f, включают в себя дополнительный флаг арифметического сброса ("arith\_reset\_flag"), который используется для выборочного сброса контекста, как было описано выше. Кроме того, арифметически кодированные спектральные данные включают множество блоков арифметических 40 данных ("arith\_data"), которые содержат арифметически кодированные спектральные значения. Структура блоков арифметически кодированных данных зависит от числа частотных полос (представленных переменной "num\_bands"), а также от состояния флага арифметического сброса, что будет рассматриваться далее.

45 Структура арифметически кодированных блоков данных будет описана со ссылкой на фиг. 6g, которая показывает синтаксическое представление указанных блоков арифметически кодированных данных. Представление данных в арифметически кодированных блоках данных зависит от числа lg кодируемых спектральных значений, статуса флага арифметического сброса, а также от контекста, то есть ранее

кодированных спектральных значений.

Контекст для кодирования текущего набора (т.е. кортежа из 2-х элементов) спектральных значений определяется в соответствии с алгоритмом определения контекста, который показан номером 660. Подробности относительно алгоритма определения контекста были рассмотрены выше (фиг. 5а и 5b). Блок арифметически кодированных данных включает в себя наборы  $\lg/2$  кодовых слов, каждый набор кодовых слов представляет множество (например, кортеж-двойку) спектральных значений. Набор кодовых слов включает в себя арифметическое кодовое слово "acod\_m [rki][m]", представляющее собой значение наиболее значимой битовой плоскости m кортежа спектральных значений, который использует от 1 до 20 бит. Кроме того, набор кодовых слов включает одно или более кодовых слов "acod\_r[r]", если кортеж спектральных значений требует больше битовых плоскостей, чем более значимая битовая плоскость, для корректной репрезентации. Кодовое слово "acod\_r[r]" представляет собой менее значимую битовую плоскость, которая использует от 1 до 20 бит.

Однако, если требуется одна или более менее значимых битовых плоскостей (в дополнение к более значимым битовым плоскостям) для правильного представления спектральных значений, то это передается с помощью одного или более арифметических кодовых слов перехода ("ARITH\_ESCAPE"). Таким образом, в целом можно сказать, что для спектрального значения определяется, сколько требуется битовых плоскостей (наиболее значимая битовая плоскость и, возможно, одна или более дополнительных менее значимых битовых плоскостей). Если требуется одна или больше менее значимых битовых плоскостей, то это передается одним или более арифметическими кодовыми словами перехода "acod\_m [rki][ARITH\_ESCAPE]", которые кодируются в соответствии с текущей выбранной сводной таблицей частот, индекс которой задается переменной rki. Кроме того, контекст адаптируется, как можно увидеть на ссылках 664, 662, если одно или более арифметических кодовых слов перехода включены в битовый поток. Следуя за одним или несколькими арифметическими кодовыми словами перехода, арифметическое кодовое слово "acod\_m [rki][m]" включается в битовый поток, как показано на ссылке 663, где rki определяет текущий действующий индекс вероятностной модели (учитывая адаптацию контекста, вызванную включением арифметических кодовых слов перехода), и где m обозначает значение наиболее значимой битовой плоскости кодируемого или декодируемого спектрального значения.

Как уже говорилось выше, присутствие любой менее значимой битовой плоскости приводит к наличию одного или более кодовых слов "acod\_r[r]", каждое из которых представляет один бит наименее значимой битовой плоскости первого спектрального значения, а также один бит наименее значимой битовой плоскости второго спектрального значения. Одно или более кодовых слов "acod\_r[r]" кодируется в соответствии с соответствующей сводной таблицей частот, которая является постоянной и независимой от контекста. Однако возможны разные механизмы для выбора сводной таблицы частот для декодирования одного или более кодовых слов "acod\_r[r]".

Кроме того, следует отметить, что контекст обновляется после кодирования каждого кортежа спектральных значений, как показано на ссылке 668, так что контекст, как правило, различен для кодирования и декодирования двух последующих кортежей спектральных значений.

Фиг. 6i показывает условные обозначения определений и вспомогательных элементов, определяющих синтаксис арифметически кодированного блока данных.

Кроме этого на фиг. 6h показан альтернативный синтаксис арифметических данных

"arith\_data()" совместно с соответствующими условными обозначениями определений и вспомогательных элементов, которые показаны на фиг. бj.

Подводя итог вышесказанному, был описан формат битового потока, который может быть обеспечен аудио кодером 100 и который может быть оценен аудио декодером 200. Битовый поток арифметически кодированных спектральных значений кодируется так, что он подходит для алгоритма декодирования, который описан выше.

Кроме того, в целом следует отметить, что кодирование является обратной операцией декодирования, так что в целом можно предположить, что кодер выполняет поиск в таблице, используя рассмотренные выше таблицы, что примерно обратно поиску в таблице, выполняемому декодером. В общем, можно сказать, что специалист в данной области, который знает алгоритм декодирования и/или синтаксис желаемого битового потока, с легкостью сможет разработать арифметический кодер, который обеспечивает данные, определенные в синтаксисе битового потока и требуемые арифметическим декодером.

Кроме этого, необходимо отметить, что механизмы для определения числового значения текущего контекста и извлечения индекса правила отображения могут быть одинаковыми в аудио кодере и аудио декодере, так как обычно необходимо, чтобы аудио декодер использовал тот же самый контекст, что и аудио кодер, так что декодирование адаптируется к процессу кодирования.

#### 15. Альтернативные варианты использования

Хотя некоторые аспекты уже были описаны в контексте устройства, ясно, что эти аспекты также представляют собой описание соответствующего способа, где блок или устройство соответствуют шагу способа или свойству шага способа. Аналогично, аспекты, изложенные в контексте шага способа, также представляют собой описание соответствующего блока или элемента либо свойства соответствующего устройства. Некоторые или все шаги способа могут быть выполнены посредством (или с помощью) аппаратного обеспечения, как, например, микропроцессор, программируемый компьютер или электронная схема. В некоторых вариантах один или несколько наиболее важных шагов способа могут быть выполнены таким устройством.

Изобретенный кодированный аудио сигнал может быть сохранен на цифровом носителе или может быть передан с помощью передающего средства, такого как беспроводное средство передачи или проводное средство передачи, например Интернет.

В зависимости от требований к определенным реализациям изобретения, воплощения изобретения могут быть реализованы в виде аппаратного средства или программного средства. Воплощение может быть осуществлено с помощью цифрового носителя, например дискеты, DVD, Blue-Ray, CD, ROM, PROM, EPROM, EEPROM или флэш-памяти, имеющего сохраненные на нем электронночитаемые контролирующие сигналы, которые взаимодействуют (или способны взаимодействовать) с программируемой компьютерной системой таким образом, что выполняется соответствующий способ. Таким образом, цифровой носитель может быть читаемым на компьютере.

Некоторые воплощения в соответствии с изобретением содержат носитель данных, имеющий электронно читаемые контролирующие сигналы, которые способны взаимодействовать с программируемой компьютерной системой так, что выполняется один из способов, описанных в данном документе.

Как правило, варианты осуществления настоящего изобретения могут быть реализованы в виде программного продукта с программным кодом, который задействован для осуществления одного из способов, когда программный продукт запускается на компьютере. Программный код, например, может быть сохранен на

считываемом носителе.

Другие варианты включают компьютерную программу, которая хранится на считываемом носителе, для выполнения одного из способов, описанных в данном документе.

5 Иными словами, воплощением изобретенного способа, следовательно, является компьютерная программа, имеющая программный код для выполнения одного из способов, описанных в данном документе, когда компьютерная программа запускается на компьютере.

10 Еще одним вариантом реализации изобретенных способов, таким образом, является носитель данных (или цифровое средство хранения, или носитель, считываемый на компьютере), включающий записанную на нем компьютерную программу для выполнения одного из способов, описанных в данном документе. Носитель данных, цифровое средство хранения или записанное средство, как правило, является материальным и/или непереходным.

15 Еще одним вариантом реализации изобретенного способа является, таким образом, поток данных или последовательность сигналов, представляющих компьютерную программу для выполнения одного из способов, описанных в данном документе. Поток данных или последовательность сигналов, например, может быть настроена для передачи через соединение передачи данных, например, через Интернет.

20 Еще один вариант реализации изобретения включает средства обработки, например, компьютер или программируемое логическое устройство, настроенное или адаптированное для выполнения одного из способов, описанных в данном документе.

25 Еще один вариант реализации изобретения включает компьютер с установленной на нем компьютерной программой для выполнения одного из способов, описанных в данном документе.

30 Еще один вариант реализации изобретения включает устройство или систему, настроенную на передачу (например, электронным или оптическим образом) компьютерной программы принимающему устройству для выполнения одного из способов, описанных в данном документе. Принимающим устройством может быть, например, компьютер, мобильное устройство, запоминающее устройство и т.д. Устройство или система может включать, например, файловый сервер для передачи компьютерной программы принимающему устройству.

35 В некоторых вариантах реализации изобретения программируемое логическое устройство (например, программируемая вентиляционная матрица) может быть использовано для выполнения некоторых или всех функциональных возможностей способов, описанных в данном документе. В некоторых вариантах программируемая вентиляционная матрица может взаимодействовать с микропроцессором для выполнения одного из способов, описанных в данном документе. Как правило, способы предпочтительно осуществляются с помощью любого аппаратного средства.

40 Описанные выше варианты осуществления изобретения являются только иллюстрацией принципов данного изобретения. Подразумевается, что модификации и варианты конфигурации и элементов, описанных в данном документе, будут очевидны для других специалистов в данной области. Таким образом, данный документ ограничивается только областью предстоящих патентных притязаний, а не конкретными 45 деталями, представленными в виде описания и объяснения вариантов реализации изобретения в настоящем документе.

## 16. Заключение

Таким образом, варианты реализации настоящего изобретения включают один или



более следующих аспектов, при этом аспекты могут использованы как по отдельности, так и в комбинации.

а) Механизм хеширования состояния контекста

Согласно аспекту изобретения состояния в хэш-таблице рассматриваются как значимые состояния и границы групп. Это позволяет значительно сократить размер необходимых таблиц.

б) Инкрементное обновление контекста

Согласно аспекту некоторые варианты реализации изобретения включают эффективный с точки зрения вычислений способ обновления контекста. Некоторые варианты используют инкрементное обновление контекста, при котором числовое значение текущего контекста извлекается из числового значения предыдущего контекста.

в) Извлечение контекста

Согласно аспекту изобретения использование суммы двух спектральных абсолютных значений представляет собой объединение с усечением членов ряда. Это вид векторного квантования спектральных коэффициентов с усилением (противоположное конвенциональному векторному квантованию с разложением по схеме shape-gain). Он нацелен на ограничение порядка контекста в процессе передачи наиболее значимой информации из окружения.

Некоторые другие методы, применяемые в вариантах реализации настоящего изобретения, рассматриваются в неопубликованных патентных заявках РСТ EP2010/065725, РСТ EP2010/065726 и РСТ EP 2010/065727. Кроме этого, в некоторых вариантах реализации настоящего изобретения используется символ остановки. В некоторых вариантах реализации настоящего изобретения для контекста рассматриваются только беззнаковые значения.

Вышеуказанные неопубликованные международные патентные заявки раскрывают аспекты, которые используются в некоторых вариантах реализации настоящего изобретения.

Например, в некоторых вариантах реализации настоящего изобретения используется идентификация нулевой зоны. Соответственно, устанавливается так называемый «флаг малого значения» (например, 16 бит числового значения текущего контекста с).

В некоторых вариантах может использоваться вычисление контекста, зависимое от диапазона. Однако, в других вариантах зависимое от диапазона вычисление контекста может быть опущено в целях уменьшения сложности вычислений и размера таблиц.

Кроме этого, важным аспектом изобретения является хеширование контекста при помощи функции хеширования. Хеширование контекста может быть основано на концепции двух таблиц, которая рассматривается в указанных выше неопубликованных международных патентных заявках. Однако, в некоторых вариантах реализации изобретения хеширование контекста может быть адаптировано определенным образом с целью повышения эффективности вычислений. В других вариантах используется хеширование контекста как оно описано в указанных выше неопубликованных международных патентных заявках.

Необходимо отметить, что инкрементное хеширование контекста является достаточно простым и эффективным с точки зрения вычислений. Кроме этого, независимость контекста от знака значений, как это используется в некоторых вариантах реализации изобретения, способствует упрощению контекста, при этом требования к памяти остаются достаточно низкими.

В некоторых вариантах реализации изобретения применяется извлечение контекста при помощи суммы двух спектральных значений и ограничения контекста. Эти два

контекста могут быть объединены. Оба аспекта направлены на ограничение порядка контекста при передаче наиболее значимой информации из окружения.

В некоторых вариантах используется «флаг малого значения», который аналогичен идентификации группы из множества нулевых значений.

5 В некоторых вариантах реализации изобретения применяется арифметический механизм остановки. Концепция подобна использованию символа "end-of-block" в JPEG, который имеет аналогичную функцию. Однако в некоторых вариантах символ ("ARITH\_STOP") эксплицитно не включен в энтропийный кодер. Наоборот, используется комбинация уже существующих символов, которые не могли встретиться ранее,  
10 например, "ESC+0". Другими словами, аудио декодер настроен выявлять комбинацию существующих символов, которые обычно не применяются для представления числового значения, а также интерпретировать появление такой комбинации уже существующих символов как арифметическое условие остановки.

15 Вариант реализации настоящего изобретения применяет механизм хеширования контекста с использованием двух таблиц.

Таким образом, некоторые варианты реализации изобретения могут включать один или более из следующих основных аспектов:

- расширенный контекст для выявления в окружении либо нулевых областей, либо областей с малыми величинами;
- 20 - хеширование контекста;
- генерация состояния контекста: инкрементное обновление состояния контекста; и
- извлечение контекста: особое квантование значений контекста, включая суммирование величин и ограничение.

Таким образом, один из аспектов реализации настоящего изобретения заключается  
25 в инкрементном обновлении контекста. Варианты реализации изобретения включают эффективную концепцию обновления контекста, которая позволяет избежать сложных вычислений в соответствии с рабочим проектом (например, рабочим проектом 5). Наоборот, в некоторых вариантах реализации изобретения используются простые операции сдвига и логические операции. Простое обновление контекста значительно  
30 упрощает вычисление контекста.

В некоторых вариантах реализации изобретения контекст независим от знака значений (например, декодированные спектральные значения). Независимость контекста от знака значений приводит к уменьшению сложности переменной контекста. Эта концепция основана на том факте, что пренебрежение знаком в контексте не приводит  
35 к значительному ухудшению эффективности кодирования.

Согласно одному из аспектов изобретения, контекст извлекается при помощи суммы двух спектральных значений. Соответственно, значительно снижаются требования к памяти, необходимой для хранения контекста. Соответственно, использование контекстного значения, которое представляет собой сумму двух спектральных значений,  
40 может рассматриваться как предпочтительное в некоторых случаях.

В некоторых случаях ограничение контекста также является усовершенствованием. В дополнение к извлечению контекста при помощи суммы двух спектральных значений, записи контекстного массива "q" ограничиваются до максимального значения "0xF" в некоторых вариантах, что, в свою очередь, приводит к ограничению требований к  
45 памяти. Ограничение значений контекстного массива "q" имеет ряд преимуществ.

В некоторых вариантах реализации изобретения используется так называемый «флаг малого значения». При получении переменной контекста с (которая также обозначена как числовое значение текущего контекста) настраивается данный флаг, если значения

ряда записей "q[1][i-3]" - "q[1][i-1]" очень малы. Соответственно, вычисление контекста может быть выполнено с высокой эффективностью. Также может быть получено значимое контекстное значение (например, числовое значение текущего контекста).

В некоторых вариантах реализации изобретения используется арифметический механизм остановки. Механизм "ARITH\_STOP" позволяет остановить арифметическое кодирование или декодирование, если остались только нулевые значения.

Соответственно, эффективность кодирования может быть повышена при умеренных затратах в плане сложности.

Согласно одному из аспектов изобретения применяется механизм хеширования контекста с использованием двух таблиц. Выполняется отображение контекста посредством алгоритма разделения на интервалы при помощи оценки таблицы "ari\_hash\_m" совместно с последующей оценкой поиска таблицы "ari\_lookup\_m". Этот алгоритм более эффективен, чем алгоритм WD3.

Далее будут рассмотрены некоторые дополнительные детали.

Необходимо отметить, что таблицы "arith\_hash\_m[600]" и "arith\_lookup\_m[600]" являются двумя различными таблицами. Первая используется для отображения одного контекстного индекса (например, числового значения контекста) по отношению к индексу вероятностной модели (например, значение индекса правила отображения), вторая используется для отображения группы последовательных контекстов, разграниченных с помощью индексов контекста в "arith\_hash\_m[]", по отношению к одной вероятностной модели.

Необходимо также отметить, что таблица "arith\_cf\_msb[96][16]" может использоваться как альтернатива таблице "arith\_cf\_m[96][17]", несмотря на то что их размеры немного отличаются. "arith\_cf\_msb[96][16]" и "arith\_cf\_m[96][17]" могут относиться к одной и той же таблице, поскольку 17-е коэффициенты вероятностных моделей всегда являются нулевыми. Это иногда не учитывается при вычислении необходимого объема для хранения таблиц.

Итак, некоторые варианты реализации изобретения представляют новую схему бесшумного кодирования (кодирования и декодирования), которая приводит к модификациям в рабочем проекте MPEG USAC (например, в рабочем проекте 5 MPEG USAC). Указанные модификации рассматриваются в прилагаемых чертежах и соответствующем описании.

В качестве заключительного пояснения необходимо отметить, что префиксы "ari" и "arith" в названии переменных, массивов, функций и т.д. взаимозаменяемы.

#### (57) Формула изобретения

1. Аудио декодер (200, 800) для обеспечения декодированной аудио информации (212, 812) на основе кодированной аудио информации (210, 810), включающий:

арифметический декодер (230; 820) для обеспечения множества декодированных спектральных значений (232, 822) на основе арифметически кодированного представления (222; 821) спектральных значений, содержащихся в кодированной аудио информации (210; 810), и

конвертер частотной области во временную область (260, 830) для обеспечения аудио представления временной области (262; 812), используя декодированные спектральные значения (232, 822), в целях получения декодированной аудио информации (212, 812);

при этом арифметический декодер (230, 820) настроен выбирать правило отображения (297; cum\_freq []), описывающее отображение значения кода арифметически кодированного представления (222, 821) спектральных значений на код символа,

представляющий одно или более спектральное значение или по меньшей мере часть одного или более декодированного спектрального значения, в зависимости от состояния контекста, описанного с помощью числового значения текущего контекста (с); и

5 при этом арифметический декодер (230, 820) настроен определять числовое значение текущего контекста (с) в зависимости от числового значения предыдущего контекста и в зависимости от множества ранее декодированных спектральных значений;

при этом арифметический декодер настроен модифицировать числовое представление числового значения предыдущего контекста, которое описывает состояние контекста для декодирования одного или более ранее декодированного спектрального значения, 10 в зависимости от значения поддиапазона контекста ( $q[\ ]$ ), которое описывает поддиапазон контекста, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста для декодирования одного или более декодируемого спектрального значения.

2. Аудио декодер по п. 1, в котором арифметический декодер настроен обеспечивать 15 числовое представление числового значения текущего контекста таким образом, что части числового представления имеют различный числовой вес, который определяется с помощью значений поддиапазона контекста ( $q[\ ]$ ).

3. Аудио декодер по п. 1, в котором числовое представление является двоичным числовым представлением одного числового значения текущего контекста (с); и при 20 этом первое подмножество битов двоичного числового представления определяется с помощью первого значения поддиапазона контекста, соответствующего одному или более ранее декодированному спектральному значению; и при этом второе подмножество битов двоичного числового представления определяется с помощью второго значения поддиапазона контекста, соответствующего одному или более ранее 25 декодированному спектральному значению, при этом биты первого битового подмножества содержат отличный от битов второго битового подмножества числовой вес.

4. Аудио декодер по п. 1, в котором арифметический декодер настроен модифицировать поразрядное маскированное подмножество информационных битов 30 числового представления числовых значений предыдущего контекста либо вариант с битовым сдвигом числового представления числового значения предыдущего контекста в зависимости от значения поддиапазона контекста, который не был учтен для извлечения числового значения предыдущего контекста, чтобы получить числовое представление числового значения текущего контекста.

35 5. Аудио декодер по п. 1, в котором арифметический декодер настроен на выполнение битового сдвига числового представления числового значения предыдущего контекста таким образом, что числовой вес битовых подмножеств, соответствующих различным значениям поддиапазона контекста, подвергается модификации, чтобы получить 40 числовое представление числового значения текущего контекста.

6. Аудио декодер по п. 5, в котором арифметический декодер настроен на выполнение битового сдвига числового представления числового значения предыдущего контекста таким образом, что битовое подмножество, соответствующее значению поддиапазона 45 контекста, удаляется из числового представления, чтобы получить числовое представление числового значения текущего контекста.

7. Аудио декодер по п. 1, в котором арифметический декодер настроен модифицировать первое битовое подмножество двоичного числового представления числового значения предыдущего контекста или вариант с битовым сдвигом двоичного 50 числового представления числового значения предыдущего контекста в зависимости

от значения поддиапазона контекста, и оставлять без изменений второе битовое подмножество двоичного числового представления числового значения предыдущего контекста или вариант с битовым сдвигом двоичного числового представления числового значения предыдущего контекста,

5        извлекать двоичное числовое представление числового значения текущего контекста из двоичного числового представления числового значения предыдущего контекста при помощи выборочного модифицирования одного или более битового подмножества, соответствующего поддиапазонам контекста, которые учитываются для декодирования ранее декодированных спектральных значений и не учитываются для декодирования  
10        текущих спектральных значений при помощи числового значения текущего контекста.

8. Аудио декодер по п. 1, в котором арифметический декодер настроен обеспечивать числовое представление числового значения текущего контекста таким образом, что подмножество наименее значимых битов числового представления числового значения  
15        текущего контекста описывает значение поддиапазона контекста, поддиапазон контекста которого используется для декодирования спектральных значений, для которых определяется состояние контекста при помощи числового значения текущего  
20        контекста, но значение поддиапазона контекста которого не используется для декодирования спектральных значений, для которых определяется состояние контекста при помощи числового значения последующего контекста.

9. Аудио декодер по п. 1, в котором арифметический декодер настроен оценивать по меньшей мере одну таблицу, чтобы определить, идентично ли числовое значение  
25        текущего контекста табличному значению контекста, которое описано с помощью записи таблицы, или находится в пределах интервала, описанного с помощью записей таблицы, а также извлекать значение индекса правила отображения, которое описывает  
30        выбранное правило отображения в зависимости от результата оценки, по меньшей мере, одной таблицы.

10. Аудио декодер по п. 1, в котором арифметический декодер настроен контролировать, является ли сумма множества значений поддиапазона контекста  
35        меньшей или равной заданному пороговому значению суммы, а также выборочно модифицировать числовое значение текущего контекста в зависимости от результата проверки.

11. Аудио декодер по п. 10, в котором арифметический декодер настроен контролировать, является ли сумма множества значений поддиапазона контекста,  
40        значения поддиапазона контекста которой соответствуют той же временной части аудио содержания, что и одно или более декодируемое спектральное значение с использованием состояния контекста, определяемого с помощью числового значения  
45        текущего контекста, а также значения поддиапазона контекста которой соответствуют более низким частотам, чем одно или более декодируемое спектральное значение с использованием состояния контекста, определяемого с помощью числового значения  
50        текущего контекста, меньшей или равной заданному пороговому значению суммы, а также выборочно модифицировать числовое значение текущего контекста в зависимости от результата проверки.

12. Аудио декодер по п. 1, в котором арифметический декодер настроен суммировать абсолютные значения первого множества ранее декодированных спектральных  
55        значений, чтобы получить первое значение поддиапазона контекста, соответствующее первому множеству ранее декодированных спектральных значений, а также суммировать абсолютные значения второго множества ранее декодированных спектральных  
60        значений, чтобы получить второе значение поддиапазона контекста, соответствующее

второму множеству ранее декодированных спектральных значений.

13. Аудио декодер по п. 1, в котором арифметический декодер настроен ограничивать значения поддиапазона контекста таким образом, что значения поддиапазона контекста оказываются репрезентативными при использовании действительного подмножества информационных битов числового представления числового значения предыдущего контекста.

14. Аудио декодер по п. 1, в котором арифметический декодер настроен обновлять двоичное числовое представление (с) числового значения предыдущего контекста, чтобы извлечь числовое значение текущего контекста (с) из числового значения предыдущего контекста, используя следующий алгоритм:

```

с=c>>4;
if(i<i max-1)
с=c(q[0][i+1]<<12);
с=(с&0xFFF0);
if(i>0)
с=с+(q[1][i-1]);

```

при этом с является переменной, представляющей, в двоичном представлении, числовое значение предыдущего контекста до выполнения алгоритма, и представляющей, в двоичном представлении, числовое значение текущего контекста после выполнения алгоритма;

при этом ">>4" обозначает операцию «сдвиг вправо на 4 бита»;

при этом i представляет индекс частоты одного или более спектрального значения, декодируемого с помощью числового значения текущего контекста;

при этом i max обозначает общее количество индексов частоты;

при этом q[0][i+1] обозначает значение поддиапазона контекста, соответствующее одному или более ранее декодированному спектральному значению для частот выше, чем частоты одного или более спектрального значения, декодируемого с помощью числового значения текущего контекста, а также для предыдущей временной части аудио содержания;

при этом "<<12" обозначает операцию «сдвиг влево на 12 битов»;

при этом &0xFFF0 обозначает булеву операцию AND с шестнадцатеричным значением 0xFFF0; и

при этом q[1][i-1] обозначает значение поддиапазона контекста, соответствующее одному или более ранее декодированному спектральному значению, для частот ниже, чем частоты одного или более спектрального значения, декодируемого с помощью числового значения текущего контекста, а также для текущей временной части аудио содержания.

15. Аудио декодер по п. 14, в котором арифметический декодер настроен выборочно модифицировать двоичное числовое представление (с) числового значения текущего контекста при помощи увеличения с на шестнадцатеричное значение 0x10000, если (q[1][i-3]+q[1][i-2]+q[1][i-1])<5;

при этом q[1][i-3], q[1][i-2], q[1][i-1] являются значениями поддиапазона контекста, каждое из которых соответствует одному или более ранее декодированному спектральному значению, для частот ниже, чем частоты одного или более спектрального значения, декодируемого с помощью числового значения текущего контекста, а также для текущей временной части аудио содержания.

16. Аудио кодер (100, 700) для обеспечения кодированной аудио информации на основе входной аудио информации (110, 710), аудио кодер, включающий:

энергосберегающий конвертер временной области в частотную область (130, 720) для обеспечения аудио представления частотной области (132, 722) на основе представления временной области (110, 710) входной аудио информации так, что аудио представление частотной области (132, 722) включает набор спектральных значений,

5 и

арифметический кодер (170; 730) настроен кодировать спектральное значение (a) или его ранее обработанную версию, используя кодовое слово переменной длины (acod\_m, acod\_r), при этом арифметический кодер (170) настроен отображать одно или более спектральное значение (a, b) или значение (m) наиболее значимой битовой плоскости одного или более спектрального значения (a, b) на значение кода (acod\_m),

10

при этом кодированная аудио информация включает множество кодовых слов переменной длины;

при этом арифметический кодер настроен выбирать правило отображения, описывающее отображение одного или более спектрального значения, или значения наиболее значимой битовой плоскости одного или более спектрального значения на значение кода в зависимости от состояния контекста (s), которое описано с помощью числового значения текущего контекста (c); и

15

при этом арифметический кодер настроен определять числовое значение текущего контекста (c) в зависимости от числового значения предыдущего контекста и в зависимости от множества ранее кодированных спектральных значений;

20

при этом арифметический кодер настроен модифицировать числовое представление (c) числового значения предыдущего контекста, которое описывает состояние контекста для декодирования одного или более ранее кодированного спектрального значения, в зависимости от значения поддиапазона контекста, которое описывает поддиапазон контекста, чтобы получить числовое представление числового значения текущего контекста, которое описывает состояние контекста, для кодирования одного или более кодируемого спектрального значения.

25

17. Способ предоставления декодированной аудио информации на основе кодированной аудио информации, включающий:

30

предоставление множества декодированных спектральных значений на основе арифметически кодированного представления спектральных значений, содержащегося в кодированной аудио информации; и

обеспечение аудио представления временной области, используя декодированные спектральные значения, чтобы получить декодированную аудио информацию;

35

при этом предоставление множества декодированных спектральных значений включает выбор правила отображения, описывающего отображение значения кода (acod\_m; value) арифметически кодированного представления (222, 821) спектральных значений, на код символа, представляющего одно или более декодированное спектральное значение или по меньшей мере часть одного или более декодированного спектрального значения, в зависимости от состояния контекста, которое описывается с помощью числового значения текущего контекста (c); и

40

при этом числовое значение текущего контекста (c) определяется в зависимости от числового значения предыдущего контекста и в зависимости от множества ранее декодированных спектральных значений;

45

при этом модифицируется числовое представление числового значения предыдущего контекста, которое описывает состояние контекста для декодирования одного или более ранее декодированного спектрального значения, в зависимости от значения поддиапазона контекста, которое описывает поддиапазон контекста, чтобы получить

числовое представление числового значения текущего контекста, которое описывает состояние контекста для декодирования одного или более декодируемого спектрального значения.

5 18. Способ предоставления кодированной аудио информации на основе входной аудио информации, включающий:

предоставление аудио представления частотной области на основе представления входной аудио информации во временной области, используя энергосберегающее преобразование из временной области в частотную, так, что аудио представление частотной области включает набор спектральных значений; и

10 арифметическое кодирование спектрального значения, или его ранее обработанной версии, используя кодовое слово переменной длины, при этом одно или более спектральное значение или значение наиболее значимой битовой плоскости одного или более спектральных значений отображается на значение кода;

15 при этом правило отображения, описывающее отображение одного или более спектрального значения, или наиболее значимой битовой плоскости одного или более спектрального значения на значение кода выбирается в зависимости от состояния контекста, которое описано с помощью числового значения текущего контекста (с); и

20 при этом числовое значение текущего контекста (с) определяется в зависимости от числового значения предыдущего контекста и в зависимости от множества ранее кодированных спектральных значений;

при этом числовое представление числового значения предыдущего контекста, которое описывает состояние контекста для кодирования одного или более ранее кодированного спектрального значения, модифицируется в зависимости от значения поддиапазона контекста, которое описывает поддиапазон контекста, чтобы получить 25 числовое представление числового значения текущего контекста, которое описывает состояние контекста для кодирования одного или более кодируемого спектрального значения.

30 19. Машиночитаемый носитель информации с записанной на него компьютерной программой для осуществления способа по п. 17, когда программа запускается на компьютере.

20. Машиночитаемый носитель информации с записанной на него компьютерной программой для осуществления способа по п. 18, когда программа запускается на 35

35

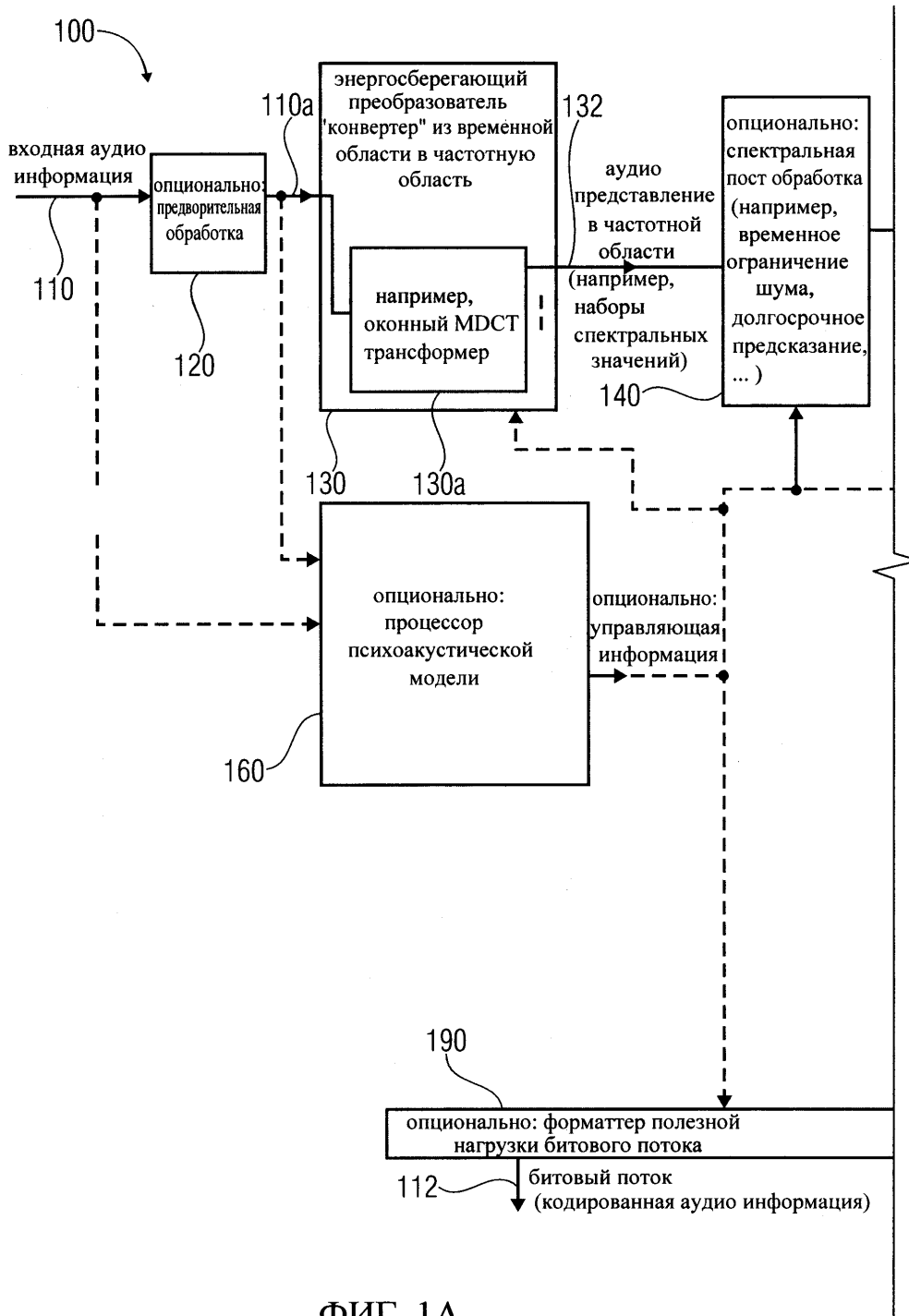
40

45



1

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

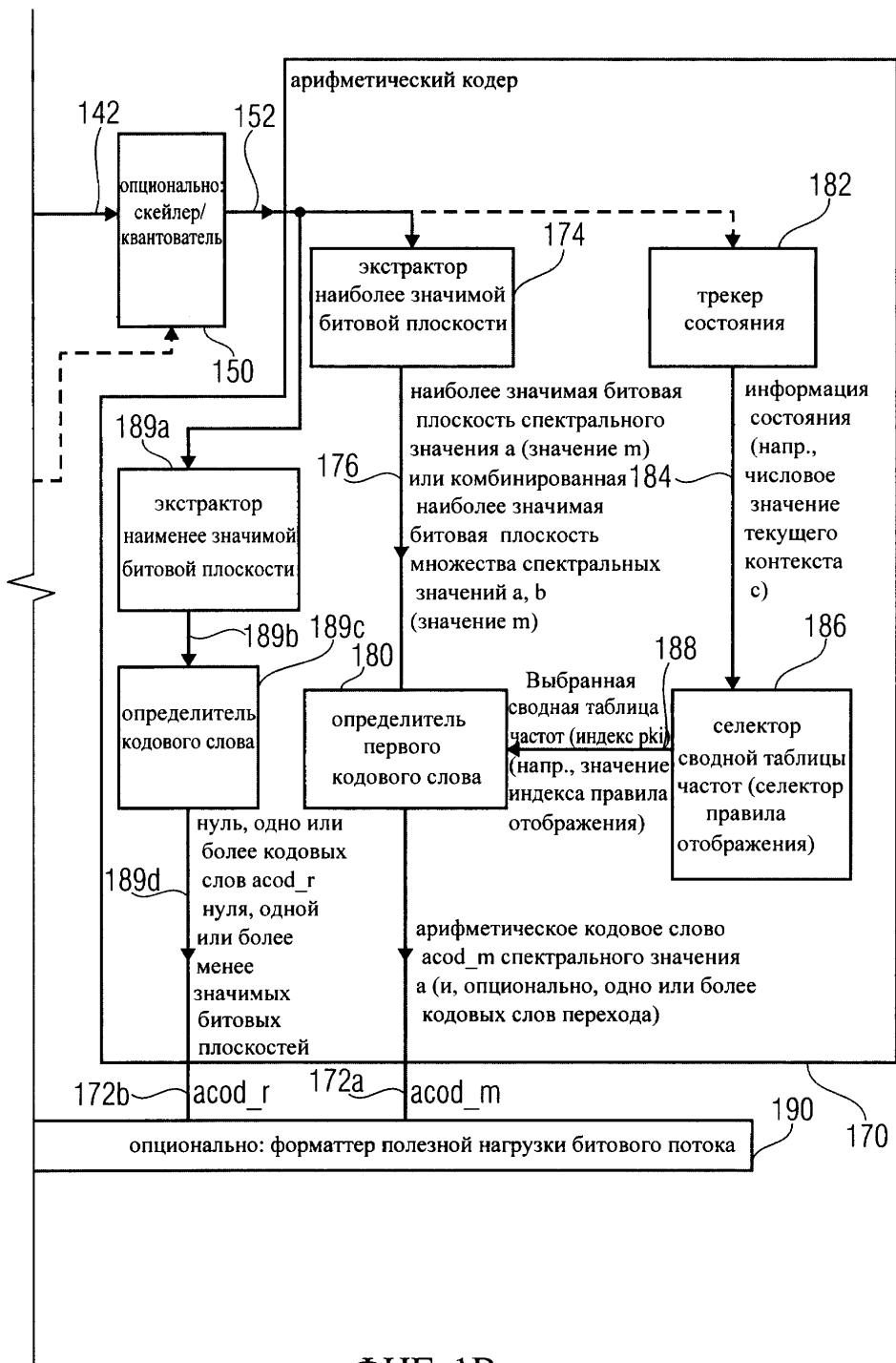


ФИГ. 1А

1

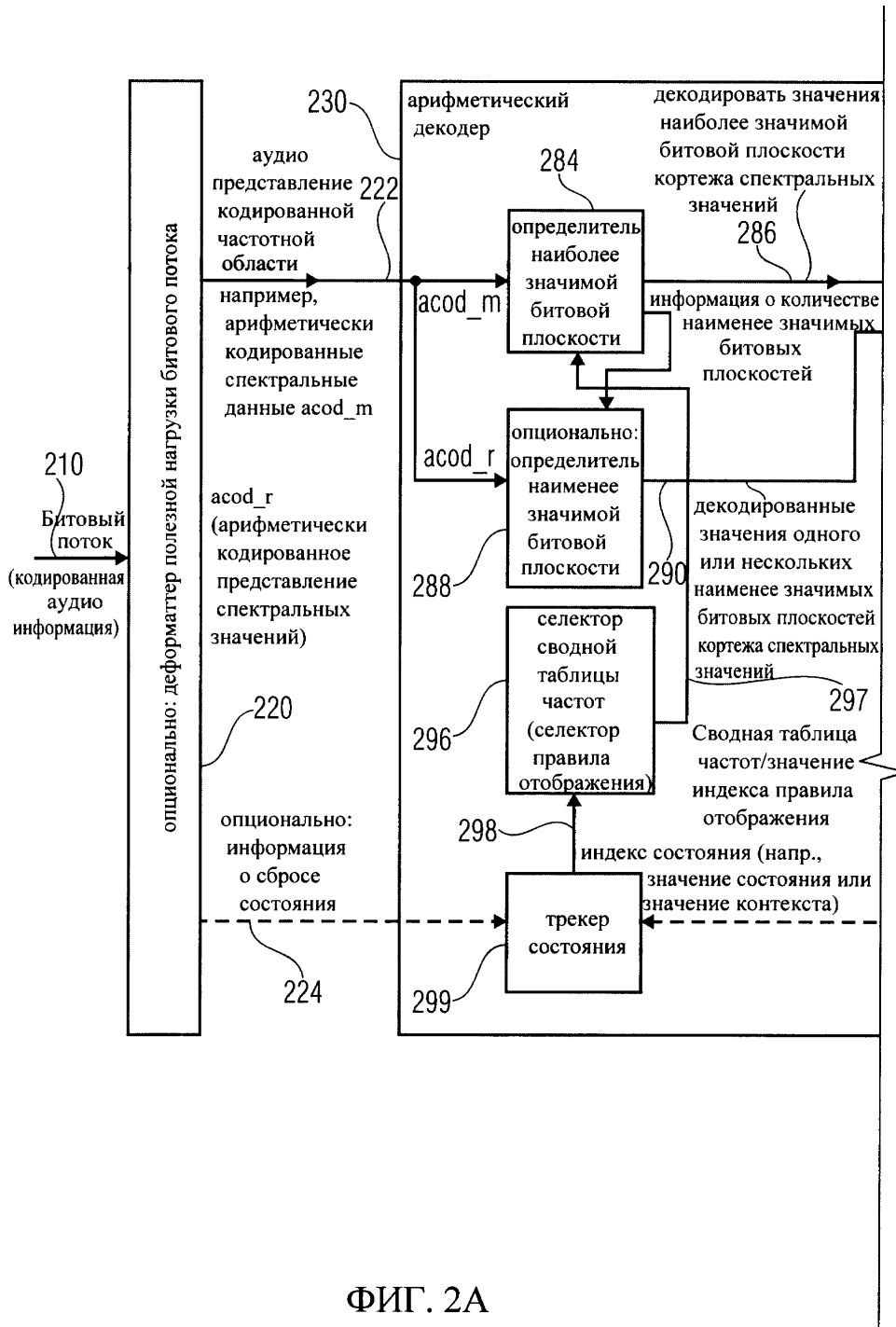
2

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



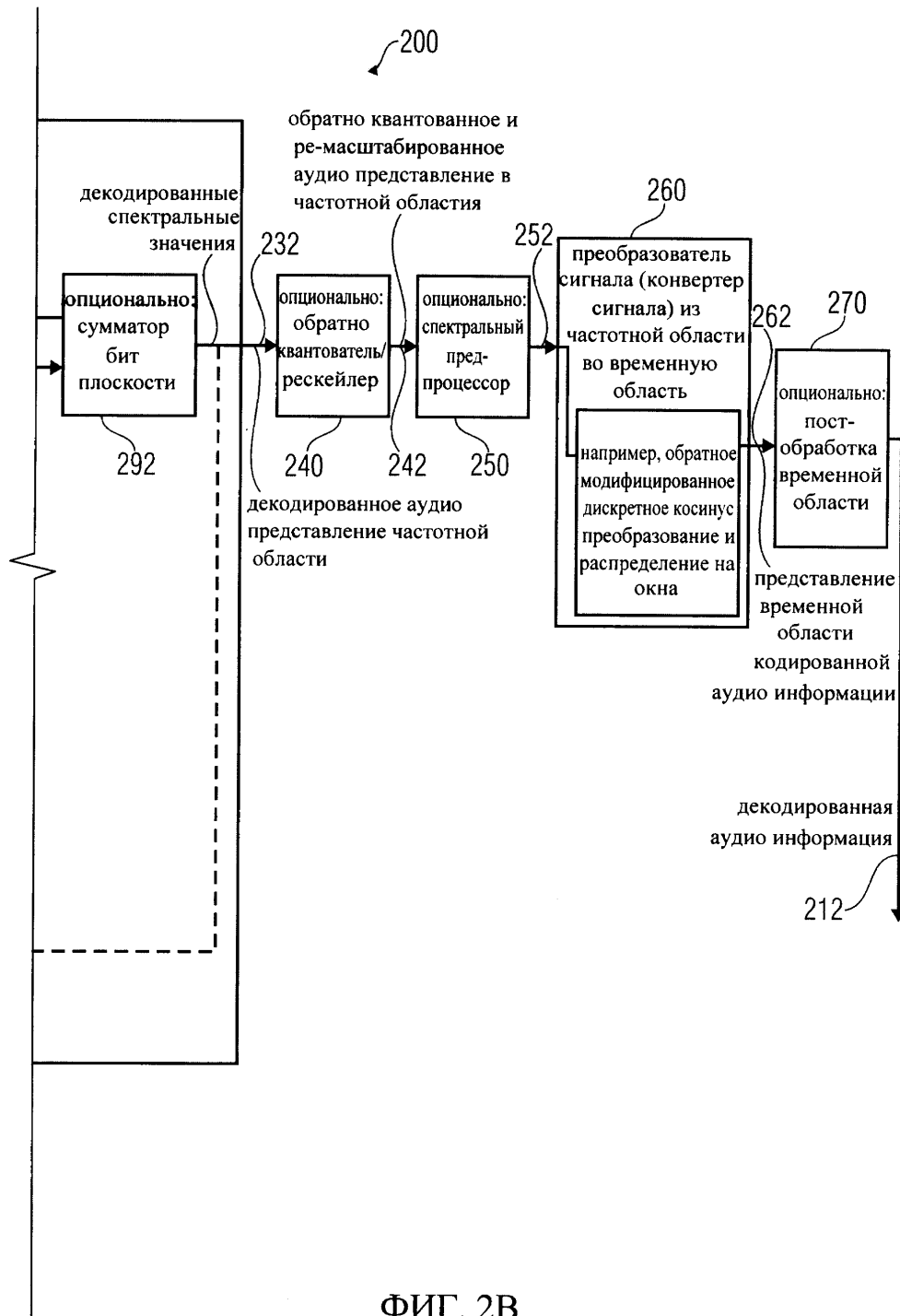
ФИГ. 1В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



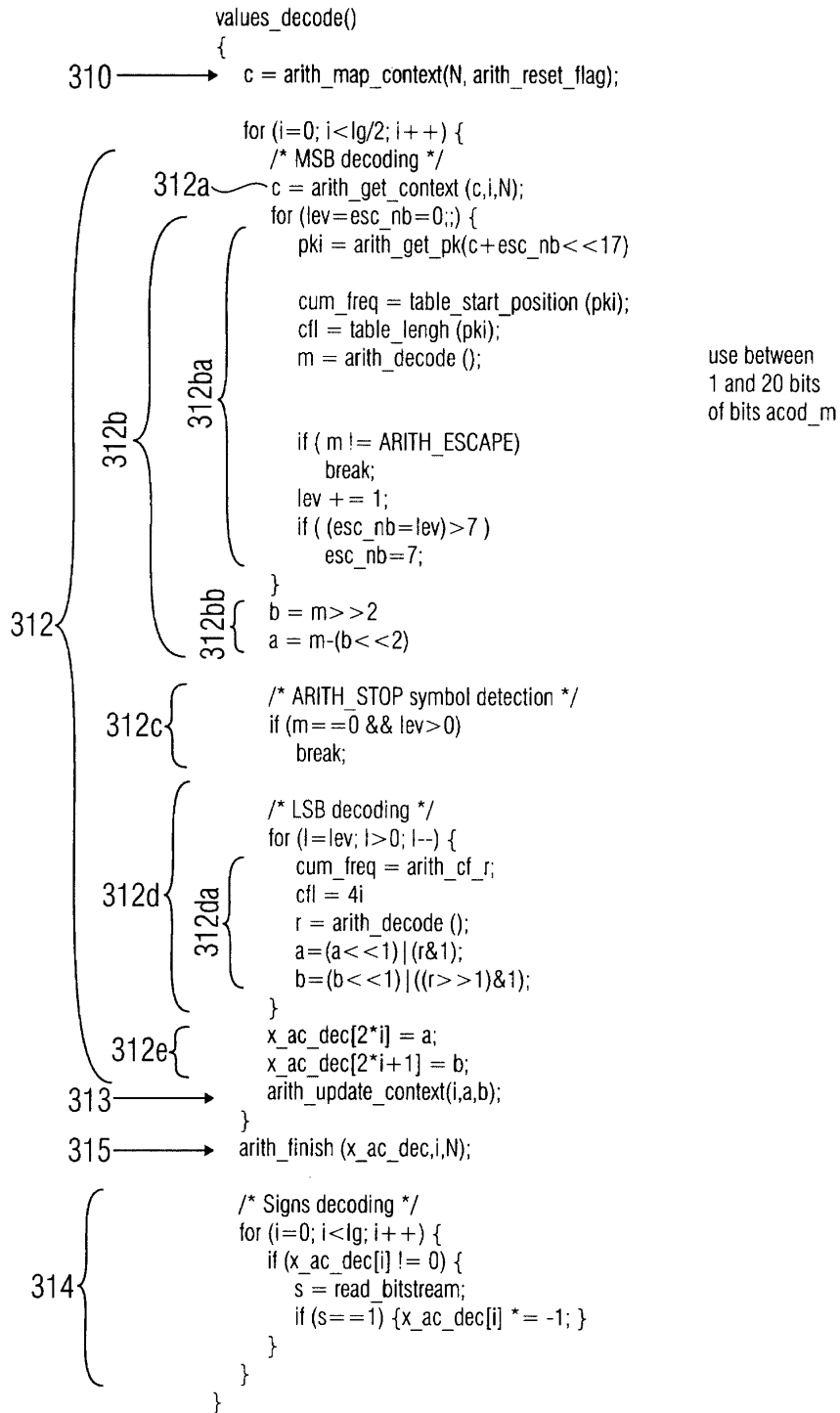
ФИГ. 2А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



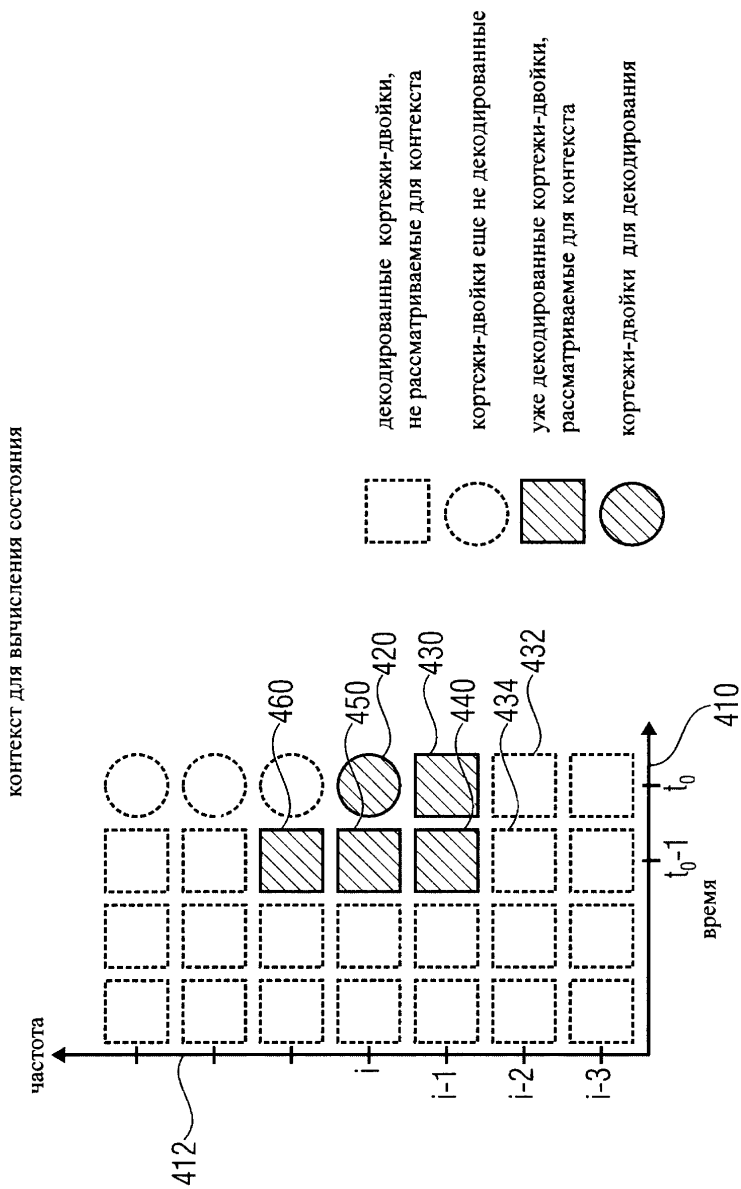
ФИГ. 2В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



ФИГ. 3

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



ФИГ. 4

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*Input variables*/
N /*Length of the current window*/
arith_reset_flag /*Arithmetic coder reset flag*/

/*Global variables*/
previous_N /*Length of the previous window */

c = arith_map_context(N,arith_reset_flag)
{
  if (arith_reset_flag) {
    for (j=0; j<N/4; j++) {
      q[0][j]=0;
    }
  } else {
    ratio = ((float)previous_N) / ((float)N);
    for (j=0; j<N/4; j++) {
      k = (int) ((float) j * ratio);
      q[0][j] = q[1][k];
    }
  }
}

previous_N=N;

return(q[0][0]<<12);
}

```

500a {

500b {

ФИГ. 5А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```
/*Input variables*/
lg /*Number of sepctral coefficients to decode in the frame*/
arith_reset_flag /*Arithmetic coder reset flag*/
/*Global variables*/
previous_lg /*Previous number of spectral lines of the previous frame*/

c=arith_map_context (lg,arith_reset_flag)
{
    v=w=0

    if(arith_reset_flag){
        for(j=0; j<lg/2; j++){
            q[0][v++] = 0;
        }
    }
    else{
        ratio= ((float)previous_lg)/((float)lg);
        for(j=0; j<lg/2; j++){
            k = (int) ((float) (j)*ratio);
            q[0][v++] = qs[w+k];
        }
    }

    previous_lg=lg;

    return(q[0][0]<<12);
}
```

ФИГ. 5В



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

504
/*Input variables*/
c /*old state context*/
i /*Index of the 2-tuple to decode in the vector*/
N /*Window Length*/

/*Output value*/
c /*updated state context*/

c = arith_get_context(c,i,N)
{
504a c = c >> 4;
      if(i < N/4-1)
504b c = c + (q[0][i+1] << 12);
504c c = (c & 0xFFF0);

      if(i > 0)
504d c = c + (q[1][i-1]);

      if (i > 3) {
504e if ((q[1][i-3] + q[1][i-2] + q[1][i-1]) < 5)
          return(c+0x10000);
      }

504f return (c);
}

```

ФИГ. 5С

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*Input variables*/
c /*old state context*/
i /*Index of the 2-tuple to decode in the vector*/
/*Output value*/
c /*updated state context*/

c=arith_get_context(c,i)
{
  c=c>>4;
  c=(c)+(q[0][i+1]<<12);
  c=(c&0xFFFO)+(q[1][i-1]);

  if(i > 3) {
    if((q[1][i-3] + q[1][i-2] + q[1][i-1]) < 5)
      return(c+0x10000);
  }

  return(c);
}

```

ФИГ. 5D

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*Input variable*/
c /*State of the context*/

/*Output value*/
pki /*Index of the probability model */

pki = arith_get_pk(c)
{
  506a {
    i_min = -1;
    i = i_min;
    i_max = (sizeof(ari_lookup_m)/sizeof(ari_lookup_m[0]))-1;
    while ((i_max-i_min)>1) {
      506b {
        506ba {
          i = i_min+((i_max-i_min)/2);
          j = ari_hash_m[i];
          if (c<(j>>8))
            i_max = i;
          else if (c>(j>>8))
            i_min=i;
          else
            return(j&0xFF);
        }
      }
    }
  }
  506c → return ari_lookup_m[i_max];
}

```

ФИГ. 5Е

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*Input variable*/
c /*State of the context*/
/*Output value*/
pki /*Index of the probability model */
/*constants*/
i_diff[]={ 299, 149, 74, 37, 18, 9, 4, 2, 1};

pki=arith_get_pk(c) {
    i_min=0;
508a { s=c<<8;
508b { for(k=0;k<9;k++) {
508ba { i=i_min+i_diff[k];
        j=ari_hash_m[i];
        if(s>j) {
            i_min=i+1;
        }
    }
}
}

j=ari_hash_m[i_min];
if(s>j)
    return(ari_lookup_m[i_min+1]);
else if(c<(j>>8))
    return(ari_lookup_m[i_min]);
else
    return(j&0xFF);
}

```

ФИГ. 5F

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*helper functions*/
bool arith_first_symbol(void);
    /* Return TRUE if it is the first symbol of the sequence,
       FALSE otherwise */
Ushort arith_get_next_bit(void);
    /* Get the next bit of the bitstream */

/* global variables */
low
high
value

/* input variables */
cum_freq[]; /* cumulative frequencies table */
cfl; /* length of cum_freq[] */

symbol = arith_decode(cum_freq, cfl)
{
    if (arith_first_symbol()) {
        value = 0;
        for (i=1; i<=16; i++) {
            value = (val<<1) | arith_get_next_bit();
        }
        low = 0;
        high = 65535;
    }
}

range = high-low+1;
cum = (((int) (value-low+1))<<14)-((int) 1))/range;
p = cum_freq-1;

```

570a {

570b {

---

ФИГ. 5G (1)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

do {
    q = p + (cfl >> 1);
    if ( *q > cum ) {p=q; cfl++; }
    cfl >>= 1;
}
while ( cfl > 1 );

symbol = p-cum_freq+1;
if (symbol)
    high = low + (range*cum_freq[symbol-1])>>14 - 1;

low += (range * cum_freq[symbol])>>14;

for (;;) {
    if (high < 32768) {}
    else if (low >= 32768) {
        value -= 32768;
        low -= 32768;
        high -= 32768;
    }
    else if (low >= 16384 && high < 49152) {
        value -= 16384;
        low -= 16384;
        high -= 16384;
    }
    else break;

    low += low;
    high += high+1;
    value = (value << 1) | arith_get_next_bit();
}
return symbol;
}

```

ФИГ. 5G (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*helper functions*/
bool arith_first_symbol(void);
    /* Return TRUE if it is the first symbol of the sequence,
    FALSE otherwise */
Ushort arith_get_next_bit(void);
    /* Get the next bit of the bitstream */

/* global variables */
low
high
value

/* input variables */
cum_freq[]; /* cumulative frequencies table */
cfl; /* length of cum_freq[] */

symbol = arith_decode(cum_freq, cfl)
{
    if (arith_first_symbol()) {
        value = 0;
        for (i=1; i<=16; i++) {
            value = (val<<1) | arith_get_next_bit();
        }
        low = 0;
        high = 65535;
    }

    range = high-low+1;
    cum = (((int) (value-low+1))<<14)-((int) 1));
    p = cum_freq-1;

    do {
        q = p + (cfl>>1);
        if ( *q *range > cum ) {p=q; cfl++; }
        cfl>>=1;
    }

```

ФИГ. 5H

15

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

while ( cfl>1 );

symbol = p-cum_freq+1;
if (symbol)
    high = low + (range*cum_freq[symbol-1])>>14 - 1;

low += (range * cum_freq[symbol])>>14;

for (;;) {
    if (high<32768) {}
    else if (low>=32768) {
        value -= 32768;
        low -= 32768;
        high -= 32768;
    }
    else if (low>=16384 && high<49152) {
        value -= 16384;
        low -= 16384;
        high -= 16384;
    }
    else break;

    low += low;
    high += high+1;
    value = (value<<1) | arith_get_next_bit();
}
return symbol;
}

```

ФИГ. 5I



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

b = m>>2;
a = m-(b<<2);
for (j=0;j<lev;j++) {
    r = arith_decode(arith_cf_r,4);
    a = (a<<1) | (r&1);
    b = (b<<1) | ((r>>1)&1);
}

```

ФИГ. 5J

```

x_ac_dec[2*i] = a
x_ac_dec[2*i+1] = b;

```

ФИГ. 5K

```

/*input variables*/
a,b /* Decoded unsigned quantized spectral coefficients of the 2-tuple */
i /* Index of the quantized spectral coefficient to decode */

arith_update_context(i, a, b)
{
    q[1][i] = a+b+1;
    if (q[1][i]>0xF)
        q[1][i] = 0xF;
}

```

ФИГ. 5L

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

/*input variables*/
offset /*number of decoded 2-tuple */
N /*Window length */
x_ac_dec /*vector of decoded spectral coefficients*/

arith_finish(x_ac_dec,offset,N)
{
  for(i=offset;i<N/4;i++) {
    x_ac_dec[2*i] = 0;
    x_ac_dec[2*i+1] = 0;
    q[1][i] = 1;
  }
}

```

ФИГ. 5М

```

b = m >> 2;
a = m & 0x03;
for(j=0;j<lev;j++){
  r = arith_decode(arith_cf_r,4);
  a = (a << 1) | (r & 1);
  b = (b << 1) | ((r > 1) & 1);
}

```

ФИГ. 5N

```

/*input variables*/
a,b /*Decoded unsigned quantized spectral coefficients of the 2-tuple*/
i /*Index of the quantized spectral coefficient to decode*/

arith_update_context () {
  qdec[2*i] = a;
  qdec[2*i+1] = b;
  q[1][i] = a + b + 1;

  if(q[1][i] > 0xF)
    q[1][i] = 0xF;
}

```

ФИГ. 5O

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```
/*input variables*/
i /*Index of the quantized spectral coefficient to decode*/
lg /*number of coefficients in the frame*/

arith_save_context(i,lg){

    for(;i<N/4;i++){
        qdec[2*i]=0;
        qdec[2*i+1]=0;
        q[1][i]=1;
    }

    if(core_mode==1){
        ratio= ((float) lg)/((float)1024);
        for(j=0; j<512; j++){
            k = (int) ((float) j*ratio);
            qs[j] = q[1][k];
        }
        previous_lg = 512;
    }
    else{
        for(j=0; j<512; j++){
            qs[j] = q[1][j];
        }
        previous_lg = MIN(1024,lg);
    }
}
```

ФИГ. 5Р

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

### Определения

a,b	Кортеж-двойка для декодирования (кортеж из двух квантованных коэффициентов для декодирования)
m	Наиболее значимая 2-битовая плоскость квантованного спектрального коэффициента для декодирования
r	Наименее значимая 2-битовая плоскость квантованного спектрального коэффициента для декодирования
lev	Уровень оставшихся битовых плоскостей Он соответствует количеству менее значимых битовых плоскостей.
arith_hash_m[]	Хэш -таблица, отображающая состояния контекста для индекса pki сводной таблицы частот
arith_lookup_m[]	Хэш таблица поиска, отображающая группу состояний контекста для индекса pki сводной таблицы частот
arith_cf_m[pki][17]	модели сводных частот для наиболее значимой 2-х битовой плоскости m и символа ARITH_ESCAPE
arith_cf_r [lsbidx][]	сводные частоты для наименее значимых битовых плоскостей символа r
arith_cf_r []	сводные частоты для наименее значимых битовых плоскостей символа r
q[2][]	Кортеж из 2-х контекстных элементов предыдущего и текущего фреймов
x_ace_dec[]	Декодированный квантованный спектральный коэффициент
arith_reset_flag	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст
ARITH_STOP	Символ остановки, состоящий из последовательности символа ARITH_ESCAPE и m=0. Когда он встречается, оставшаяся часть фрейма декодируется с нулевыми значениями.
N	Длина окна. Для режима FD он выводится из window_sequence для TCX $N=2*lg$ .
previous_N	Длина предыдущего окна.

ФИГ. 5Q

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

### Определения

a,b	Кортеж-двойка для декодирования (кортеж из двух квантованных коэффициентов для декодирования)
m	Наиболее значимая 2-битовая плоскость квантованного спектрального коэффициента для декодирования
r	Наименее значимая 2-битовая плоскость квантованного спектрального коэффициента для декодирования
lev	Уровень оставшихся битовых плоскостей Он соответствует количеству битовых плоскостей, менее значимых, чем наиболее значимые 2-х битовые плоскости.
arith_hash_m[]	Хэш -таблица, отображающая состояния контекста для индекса rki сводной таблицы частот
arith_lookup_m[]	Хэш таблица поиска, отображающая группу состояний контекста для индекса rki сводной таблицы частот
arith_cf_m[pki][17]	модели сводных частот для наиболее значимой 2-х битовой плоскости m и символа ARITH_ESCAPE
arith_cf_r []	сводные частоты для наименее значимых битовых плоскостей символа r
previous_lg	Количество переданных спектральных коэффициентов, ранее декодированных арифметическим декодером
q[2][[]]	Текущий контекст кортежей из 2-х элементов, используемых для декодирования текущего фрейма
qs[]	Прошлый контекст, сохраняемый для следующего фрейма
qdec[]	Декодированные квантованные спектральные коэффициенты
arith_reset_flag	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст
ARITH_STOP	Символ остановки, состоящий из последовательности символа ARITH_ESCAPE и m=0. Когда он встречается, оставшаяся часть фрейма декодируется с нулевыми значениями.
N	Длина окна. Для режима FD он выводится из window_sequence (см. раздел 6.8.3.1.) для TCX N=2*lg.

### ФИГ. 5R

21

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

usac_raw_data_block ()
{
    single_channel_element (); and/or
    channel_pair_element ();
}

```

ФИГ. 6А

## Syntax of single\_channel\_element()

синтаксис	количество бит	мнемоник
<pre> single_channel_element() {     <b>core_mode</b>     if ( core_mode == 1 ) {         lpd_channel_stream();     }     else {         fd_channel_stream();     } } </pre>	1	<b>uimsbf</b>

ФИГ. 6В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Syntax of channel\_pair\_element()

синтаксис	количество бит	мнемоник
channel_pair_element() {		
<b>core_mode0</b>	<b>1</b>	<b>uimsbf</b>
<b>core_mode1</b>	<b>1</b>	<b>uimsbf</b>
ics_info();	optional: common ics_info for two channels	
if ( core_mode0 == 1 ) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
if ( core_mode1 == 1 ) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
}		

ФИГ. 6С

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

**Syntax of ics\_info()**

СИНТАКСИС	КОЛИЧЕСТВО БИТ	МНЕМОНИК
<code>ics_info()</code>		
<code>{</code>		
<code>  window_length;</code>	<b>1</b>	<b>uimsbf</b>
<code>  if(window_length != 0) {</code>		
<code>    transform_length;</code>	<b>1</b>	<b>uimsbf</b>
<code>  }</code>		
<code>  else {</code>		
<code>    transform_length=0;</code>		
<code>  }</code>		
<code>  window_shape;</code>	<b>1</b>	<b>uimsbf</b>
<code>  if (window_length != 0 &amp;&amp; transform_length != 0){</code>		
<code>    max_sfb;</code>	<b>4</b>	<b>uimsbf</b>
<code>    scale_factor_grouping;</code>	<b>7</b>	<b>uimsbf</b>
<code>  }</code>		
<code>  else {</code>		
<code>    max_sfb;</code>	<b>6</b>	<b>uimsbf</b>
<code>  }</code>		
<code>}</code>		

} опционально

ФИГ. 6D



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

**Syntax of fd\_channel\_stream()**

синтаксис	количество бит	мнемоник
fd_channel_stream() { <b>global_gain;</b>	<b>8</b>	<b>uimsbf</b>
ics_info();	(unless included in channel pair element)	
scale_factor_data ();		
ac_spectral_data ();		
}		

ФИГ. 6E

**Syntax of ac\_spectral\_data()**

синтаксис	количество бит	мнемоник
ac_spectral_data() { <b>arith_reset_flag</b>	<b>1</b>	<b>uimsbf</b>
for (win=0; win<num_windows; win++){ arith_data(num_bands, arith_reset_flag)		
}		
}		

ФИГ. 6F

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

### Syntax of arith\_data()

синтаксис	количество бит	мнемоник
<pre>arith_data(lg, arith_reset_flag) {   c = arith_map_context(N, arith_reset_flag);    for (i=0; i&lt;lg/2; i++) {     /* MSB decoding */     c = arith_get_context(c,i,N);     for (lev=esc_nb=0;;) {       662 { pki = arith_get_pk(c+esc_nb&lt;&lt;17)       663 { acod_m[pki][m]       664 { if ( m != ARITH_ESCAPE)           break;           lev += 1;           if ( (esc_nb=lev)&gt;7 )             esc_nb=7;         }         b = m&gt;&gt;2;         a = m - (b&lt;&lt;2);          /* ARITH_STOP symbol detection */         if (m==0 &amp;&amp; lev&gt;0)           break;          /* LSB decoding */         for (l=lev; l&gt;0; l--) {            acod_r[r]           a=(a&lt;&lt;1) (r&amp;1);           b=(b&lt;&lt;1) ((r&gt;&gt;1)&amp;1);         }         x_ac_dec[2*i] = a;         x_ac_dec[2*i+1] = b;         668 { arith_update_context(i,a,b);       }       arith_finish(x_ac_dec,lg,N);        /* Signs decoding */       for (i=0; i&lt;lg; i++) {         if (x_ac_dec[i] != 0) {           s;           if (s==1) {x_ac_dec[i] *= -1; }         }       }     }   } }</pre>	<p>1..20</p> <p>1..20</p> <p>1</p>	<p>viclbf</p> <p>viclbf</p> <p>uimbsf</p>

ФИГ. 6G

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Таблица Syntax of arith\_data()

синтаксис	количество бит	мнемоник
<pre> Arith_data(lg, arith_reset_flag){   c=arith_map_context(lg, arith_reset_flag);   for (i=0; i&lt;lg/2; i++) {     /*MSBs decoding*/     c = arith_get_context (c,i);     for (lev=esc_nb=0;;) {       pki = arith_get_pk(c+esc_nb&lt;&lt;17)       <b>acod_m</b>[pki][m]       if ( m != ARITH_ESCAPE)         break;       lev += 1;       if((esc_nb=lev)&gt;7)         esc_nb=7;     }     b=m&gt;&gt;2;     a=m-(b&lt;&lt;2);      /*ARITH_STOP symbol detection*/     if(m==0 &amp;&amp; lev&gt;0)       break;      /*LSBs decoding*/     for (l=lev; l&gt;0; l--) {       <b>acod_r</b>[l]       a=(a&lt;&lt;1) (r&amp;1);       b=(b&lt;&lt;1) ((r&gt;&gt;1)&amp;1);     }     arith_update_context(a,b,i);   }   arith_save_arith (l,lg);    /*Signs decoding*/   for (i=0; i&lt;lg/2; i++) {     if(a!=0){       s;       if(s) a=-a;     }     if(b!=0){       s;       if(s) b=-b;     }   } } </pre>	<p>1..20</p> <p>1..20</p> <p>1</p> <p>1</p>	<p>vclbfb</p> <p>vclbfb</p> <p>uimbsfb</p> <p>uimbsfb</p>

ФИГ. 6H

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

### Определения

<code>arith_data()</code>	элемент данных для декодирования данных спектрального бесшумного кодера
<code>arith_reset_flag</code>	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст
<code>acod_m[<i>pk</i>][<i>m</i>]</code>	Арифметическое кодовое слово, необходимое для декодирования наиболее значимой 2-битовой плоскости <i>m</i> кортежа из 2-х квантованных спектральных коэффициентов
<code>acod_r[<i>lsbidx</i>][<i>r</i>]</code>	Арифметическое кодовое слово, необходимое для декодирования остаточных битовых плоскостей <i>r</i> кортежа из 2-х квантованных спектральных коэффициентов
<code>s</code>	Кодированный знак ненулевого спектрального квантованного коэффициента

### Вспомогательные элементы

<code>a,b</code>	Кортеж-двойка, соответствующая квантованным спектральным коэффициентам
<code>m</code>	Декодируемая наиболее значимая 2-битовая плоскость кортежа из 2-х элементов
<code>r</code>	Декодируемая наименее значимые плоскости кортежа из 2-х элементов
<code>lg</code>	количество квантованных коэффициентов для декодирования
<code>N</code>	длина окна Для AAC выводится из <code>window_sequence</code> и для TCX $N=2*lg$ .
<code>i</code>	индекс декодируемого кортежа из 2-х элементов в пределах фрейма
<code>pk</code>	индекс сводной таблицы частот, используемый арифметическим декодером для декодирования <i>m</i> .
<code>arith_get_pk ()</code>	Функция, которая возвращает индекс <i>pk</i> сводной таблицы частот, необходимый для декодирования кодового слова <code>acod_m[<i>pk</i>][<i>a</i>]</code> .
<code>c</code>	состояние контекста
<code>lsbidx</code>	индекс сводной таблицы частот, используемый арифметическим декодером для декодирования <i>r</i>
<code>lev</code>	Уровень декодируемых битовых плоскостей кроме наиболее значимой 2-х битовой плоскости
<code>ARITH_ESCAPE</code>	Символ перехода, который указывает дополнительные декодируемые плоскости, кроме двух наиболее значимых битовых плоскостей.
<code>esc_nb</code>	Число символов <code>ARITH_ESCAPE</code> , уже декодированных для текущего кортежа из 2-х элементов. Значение ограничено 7.
<code>x_ac_dec[]</code>	Элемент, содержащий декодированные спектральные коэффициенты
<code>arith_map_context()</code>	Инициализирует контексты, необходимые для декодирования текущего фрейма
<code>arith_get_context()</code>	Вычисляет состояние контекста для декодирования текущего кортежа из 2-х символов <i>m</i>
<code>arith_update_context()</code>	Обновляет кортеж для следующего кортежа
<code>arith_finish ()</code>	Закончить бесшумное кодирование

ФИГ. 6I

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

### Определения

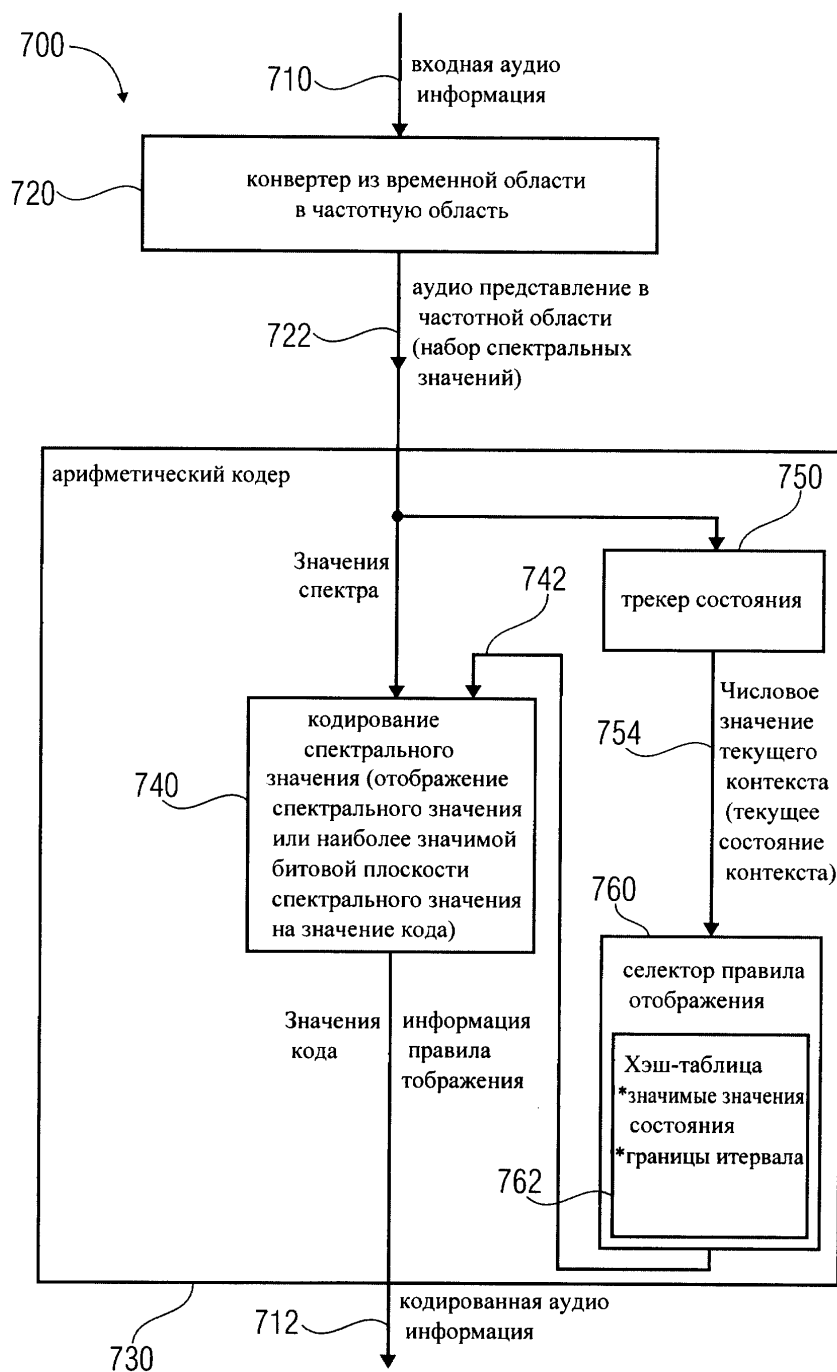
<code>arith_data()</code>	элемент данных для декодирования данных спектрального бесшумного кодера
<code>arith_reset_flag</code>	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст
<code>acod_m[pki][m]</code>	Арифметическое кодовое слово, необходимое для декодирования наиболее значимой 2-битовой плоскости <i>m</i> кортежа из 2-х квантованных спектральных коэффициентов
<code>arith_r[]</code>	необходимое для декодирования остаточных битовых плоскостей <i>r</i> кортежа из 2-х квантованных спектральных коэффициентов
<code>s</code>	Кодированный знак ненулевого спектрального квантованного коэффициента

### Вспомогательные элементы

<code>a,b</code>	Кортеж-двойка, соответствующая квантованным спектральным коэффициентам
<code>m</code>	Декодируемая наиболее значимая 2-битовая плоскость кортежа из 2-х элементов
<code>r</code>	Декодируемая наименее значимые плоскости кортежа из 2-х элементов
<code>lg</code>	количество квантованных коэффициентов для декодирования
<code>i</code>	индекс декодируемого кортежа из 2-х элементов в пределах фрейма
<code>pki</code>	индекс сводной таблицы частот, используемый арифметическим декодером для декодирования <i>m</i> .
<code>arith_get_pk ()</code>	Функция, которая возвращает индекс <i>pki</i> сводной таблицы частот, необходимый для декодирования кодового слова <code>acod_m[pki][a]</code> .
<code>c</code>	состояние контекста
<code>lev</code>	Уровень декодируемых битовых плоскостей кроме наиболее значимой 2-х битовой плоскости
<code>ARITH_ESCAPE</code>	Символ перехода, который указывает дополнительные декодируемые плоскости, кроме двух наиболее значимых битовых плоскостей.
<code>esc_nb</code>	Число символов <code>ARITH_ESCAPE</code> , уже декодированных для текущего кортежа из 2-х элементов. Значение ограничено 7.
<code>arith_map_context()</code>	Инициализирует контексты, необходимые для декодирования текущего фрейма
<code>arith_get_context()</code>	Вычисляет состояние контекста для декодирования текущего кортежа из 2-х символов <i>m</i>
<code>arith_update_context()</code>	Обновляет кортеж для следующего кортежа
<code>arith_save_context()</code>	Сохранить контекст для следующего декодируемого фрейма

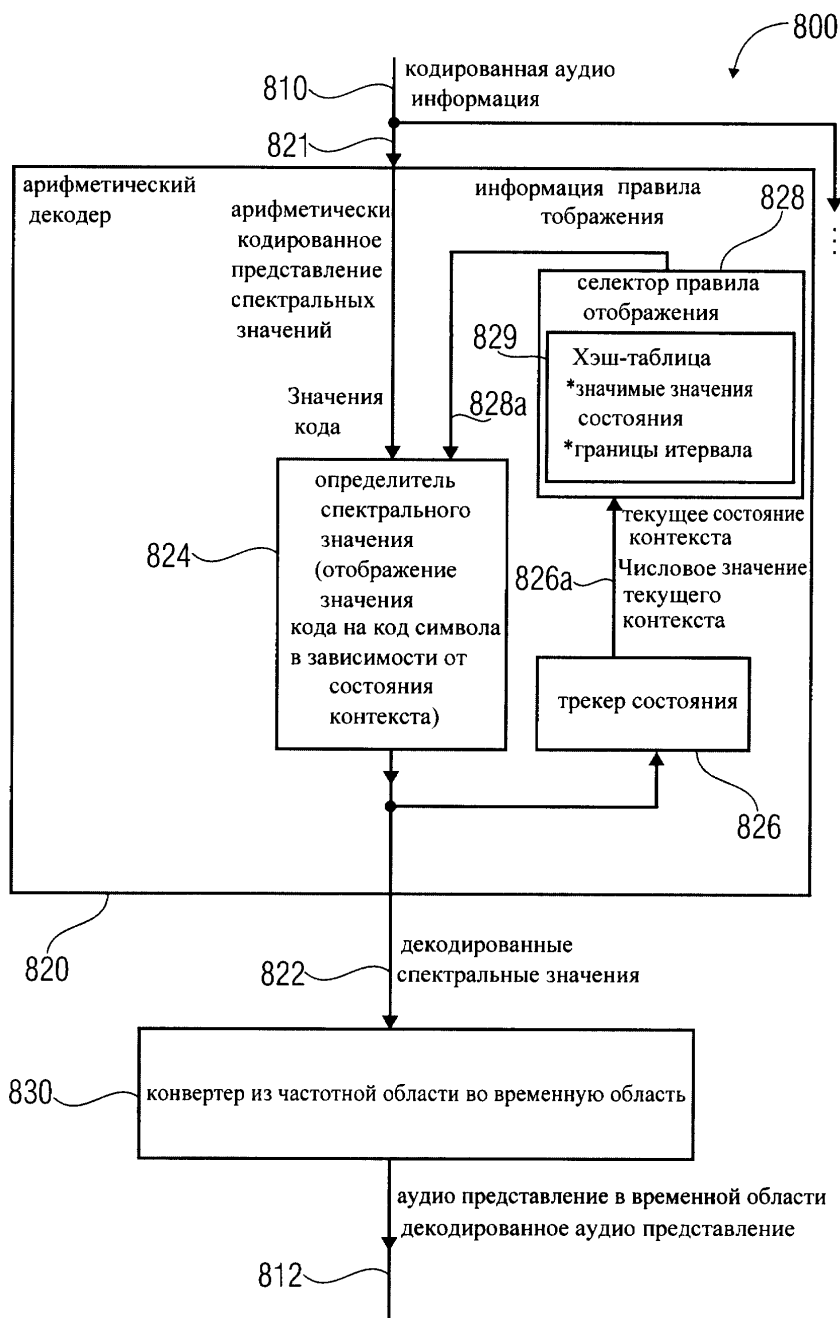
## ФИГ. 6J

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



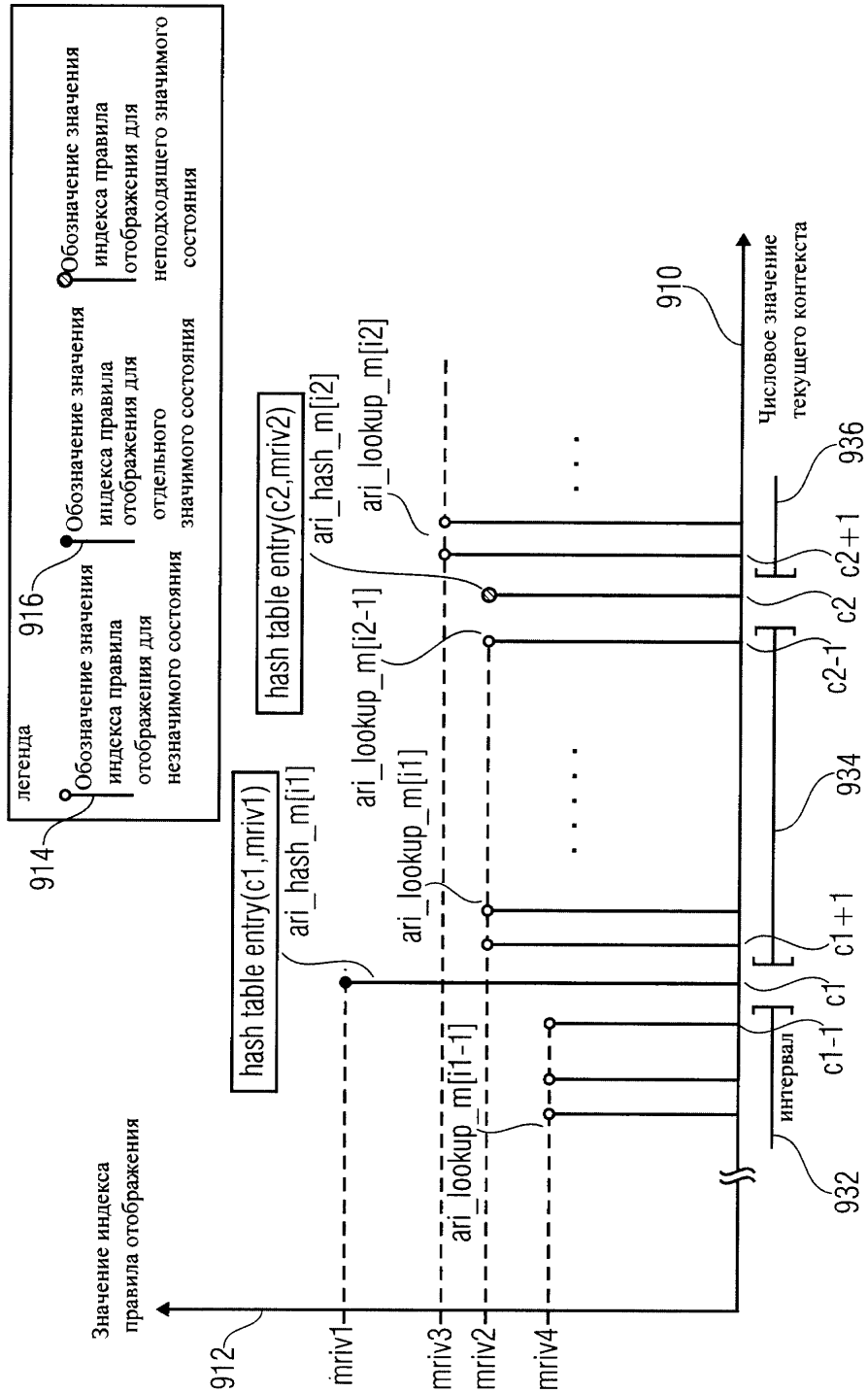
ФИГ. 7

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



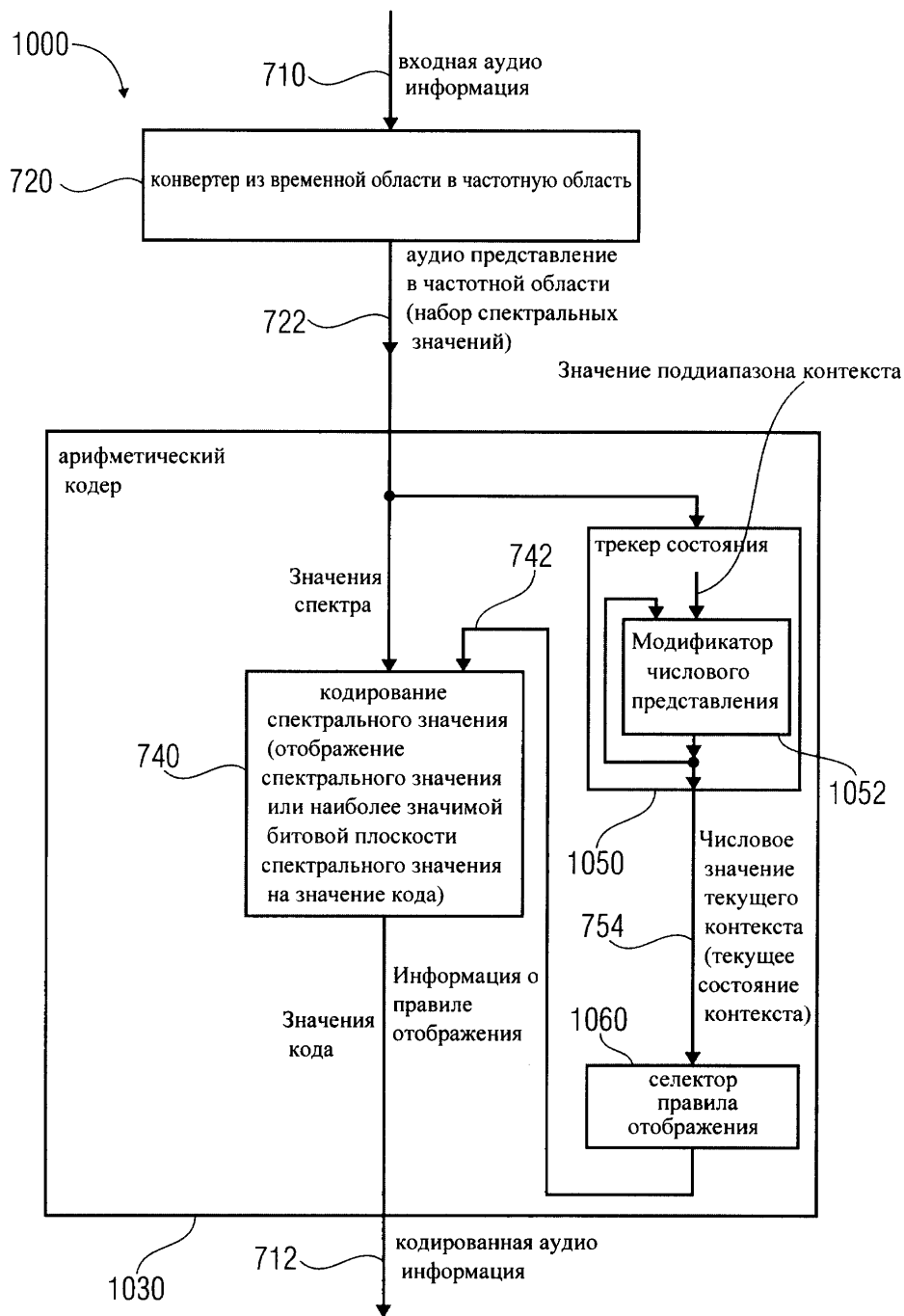
ФИГ. 8

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



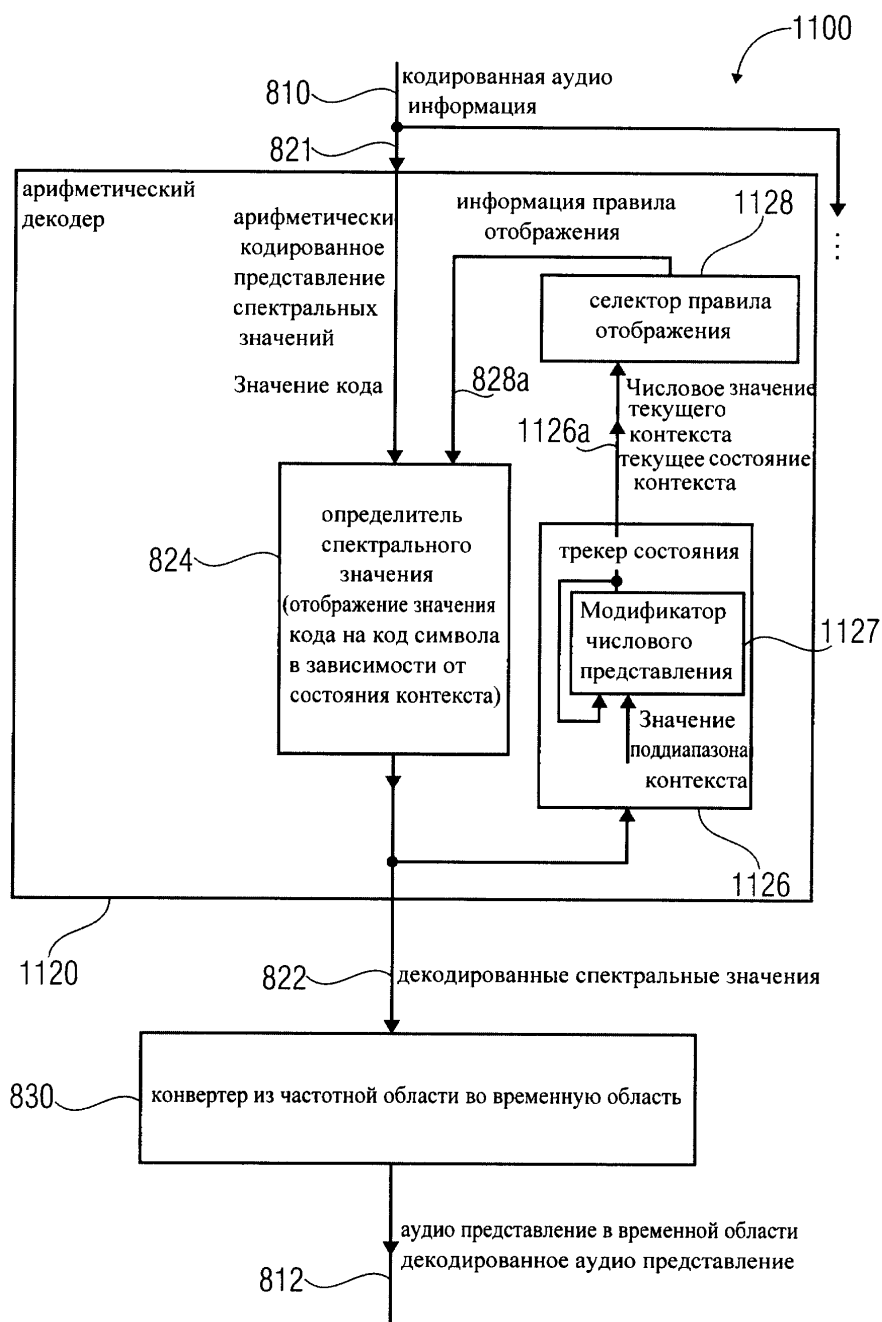


Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



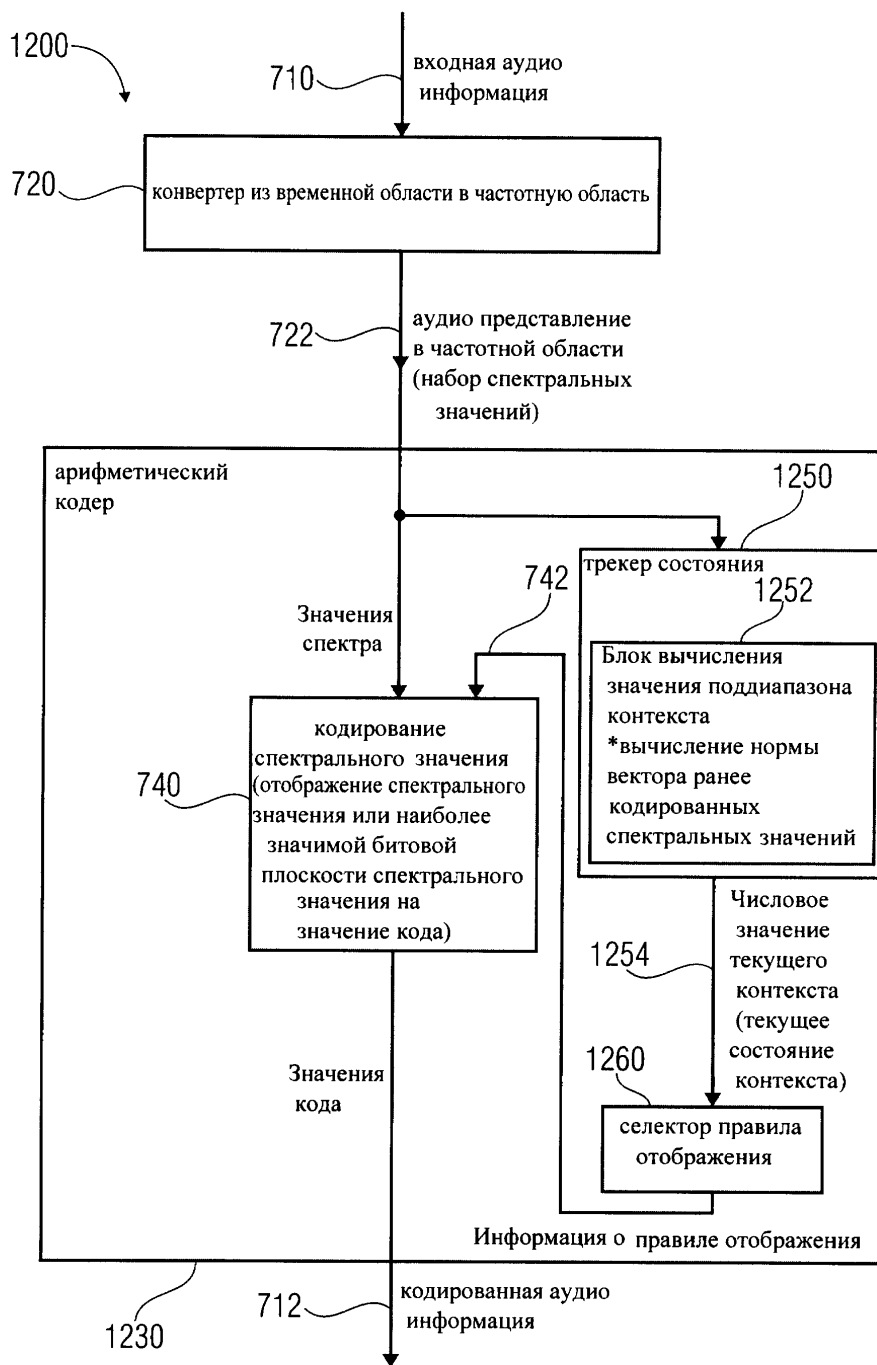
ФИГ. 10

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



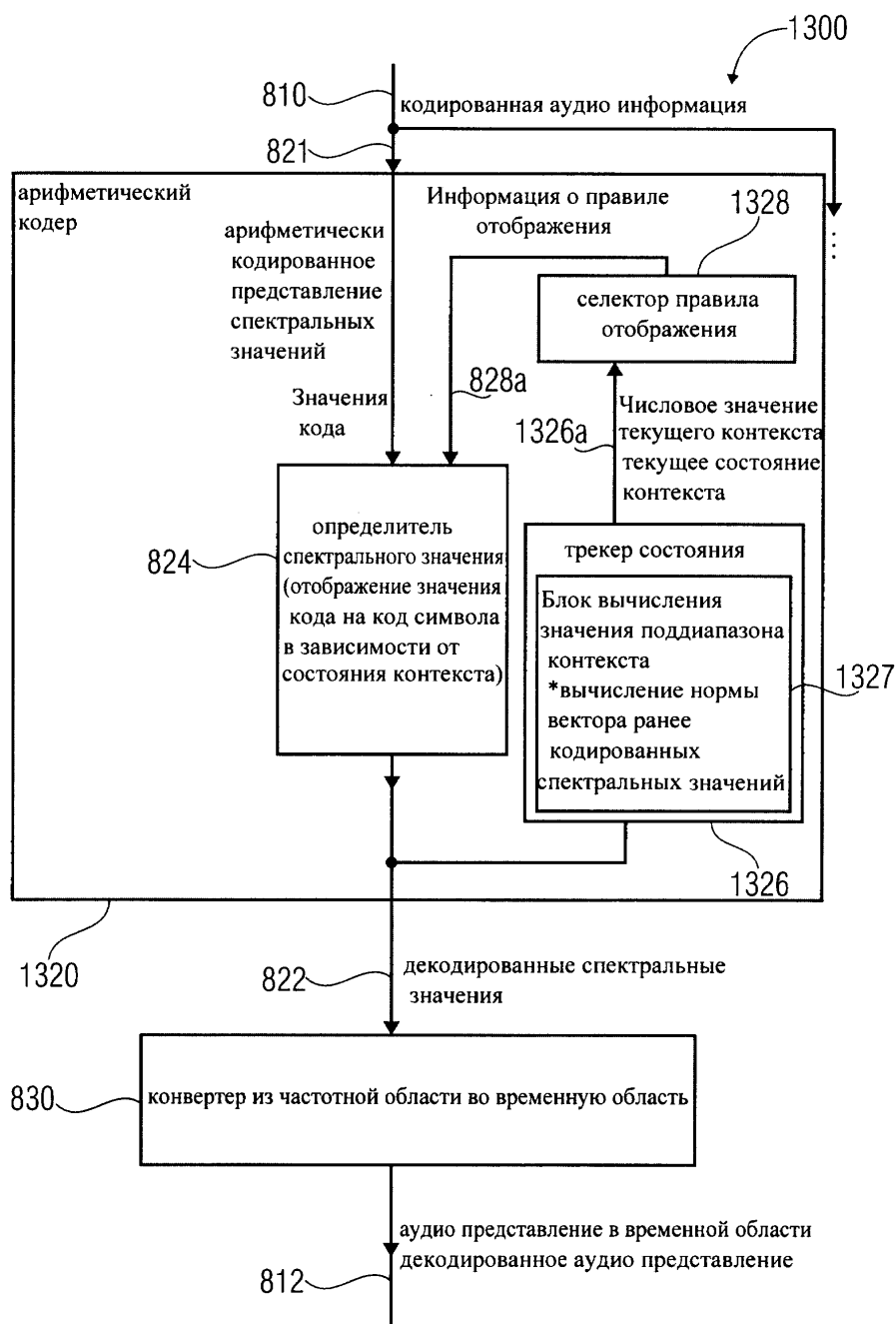
ФИГ. 11

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



ФИГ. 12

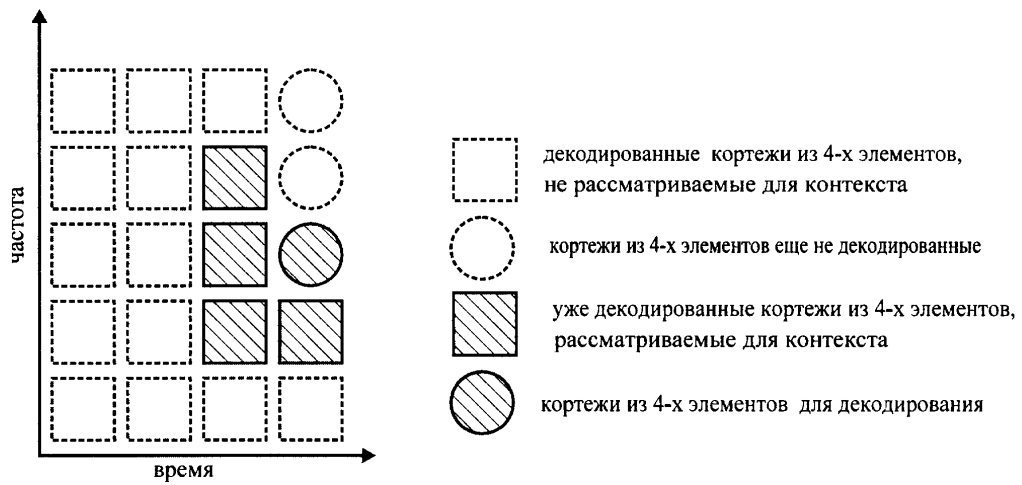
Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста



ФИГ. 13

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Контекст для вычисления состояния,  
используемый в WD4 USAC



ФИГ. 14А

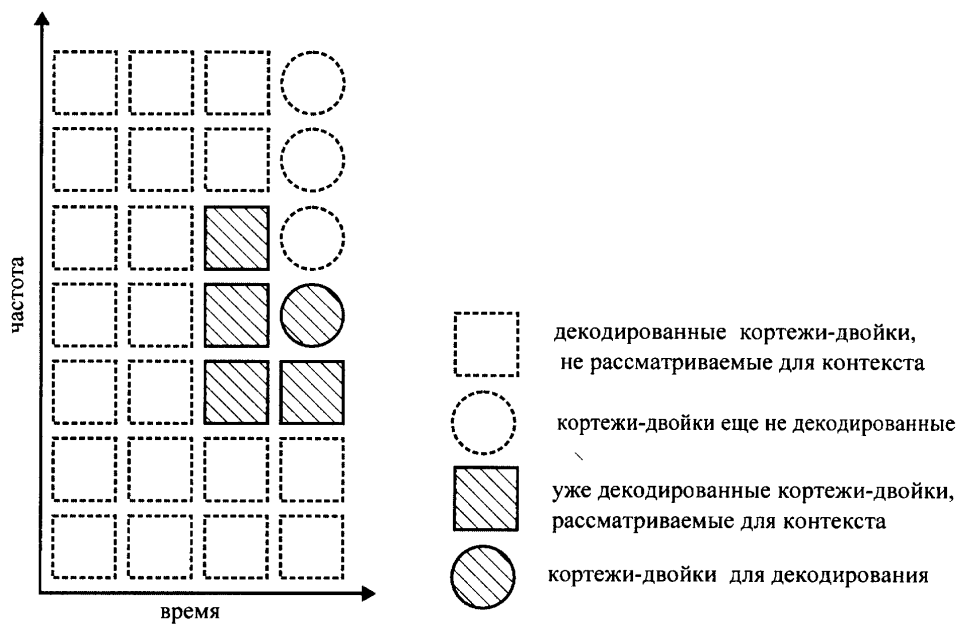
Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Название таблицы	описание	Единица данных	память (слова из 32 бит)
arith_cf_ng_hash[128]	Хэш-таблица, отображающая контекст на индекс модели вероятности	word	128
arith_cf_ng[32][545]	сводные частоты групп для каждого режима распределения вероятности,	1/2 word	8720
egroups[8][8][8][8]	индекс группы для кортежа из 4-х элементов	1/2 word	2048
dgvectors[4*4096]	индекс группы отображения и индекс элемента для кортежа из 4-х элементов	1/4 word	4096
dgroups[544]	индекс группы отображения для кардинального числа группы и смещение в dg векторах	word	544
arith_cf_ne[2701]	сводные частоты символа индекса элемента	1/2 word	1350.5
arith_cf_r[16]	сводные частоты наименее значимых битовых плоскостей	1/2 word	8
итого:			<b>16894.5</b>

ФИГ. 14В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Контекст для вычисления состояния,  
используемый в предложенной схеме



ФИГ. 15А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Таблицы, используемые в предложенной схеме кодирования

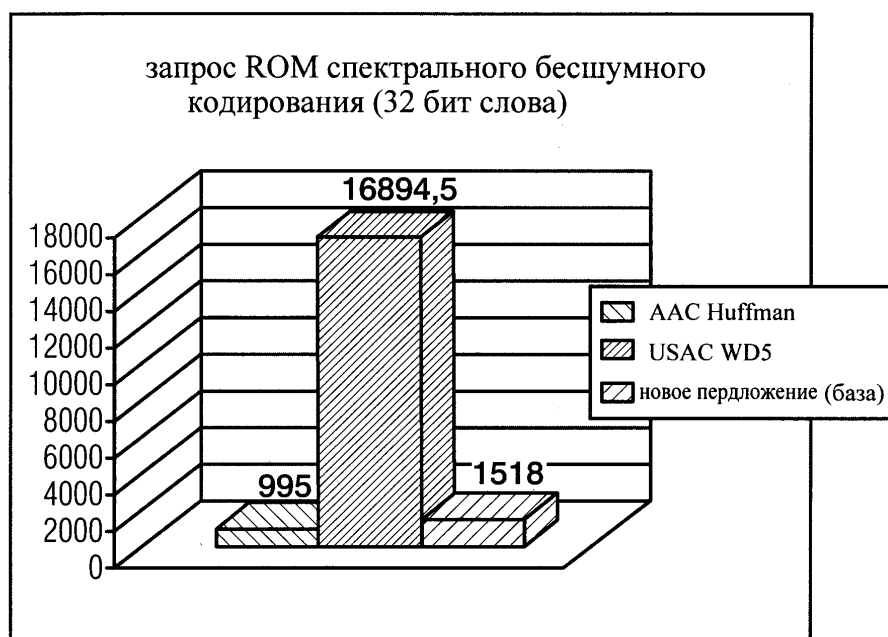
Название таблицы	описание	Единица данных	память (слова из 32 бит)
arith_hash[600]	Хэш-таблица, отображающая состояния контекст на группу состояний	1 word	600
arith_lookup[600]	Отображение таблицы поиска	1/4 word	150
arith_cf_msb[96][16]	Группы состояний моделей таблицы сводных частот для наиболее значимой 2-х битовой плоскости m и символа ARITH_ESCAPE	1/2 word	768
Итого:			<b>1518</b>

ФИГ. 15В



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

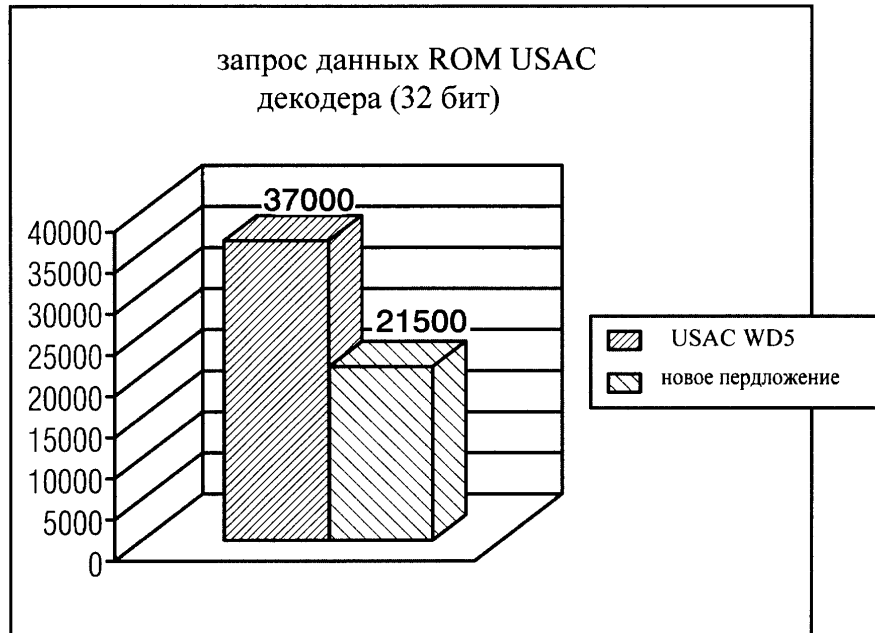
запрос ROM схемы бесшумного кодирования,  
как предлагается и в WD4



ФИГ. 16А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

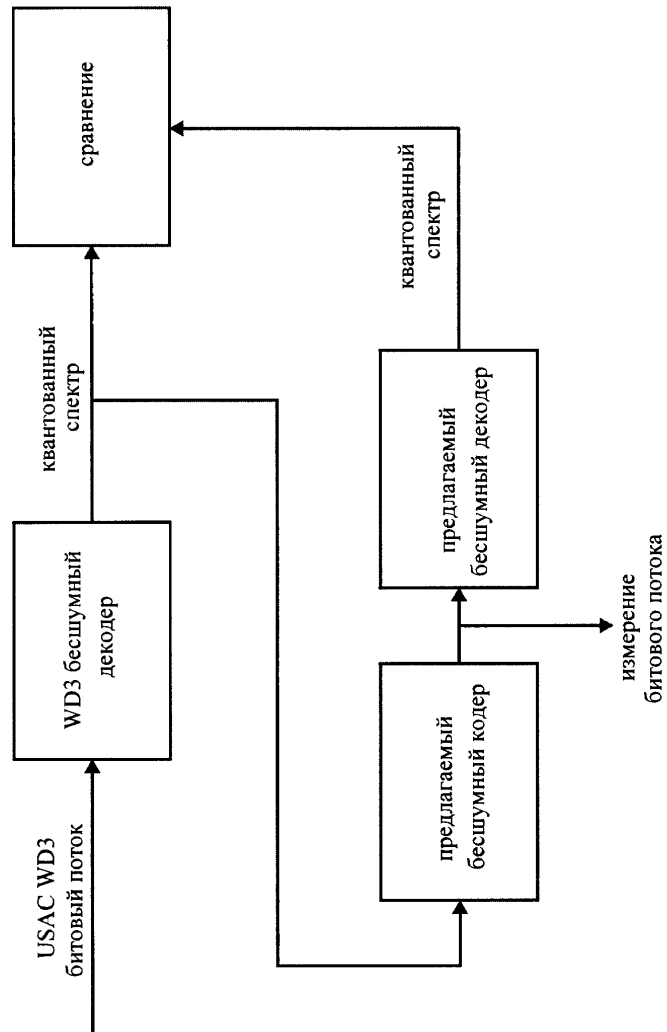
общий запрос данных ROM USAC декодера,  
WD4 и текущее предложение



ФИГ. 16В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

сравнение WD3 бесшумного кодирования с предлагаемой схемой кодирования



ФИГ. 17

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Средние битрейты, производимые арифметическим кодером WD3 и новым предложением. Чистый битрейт не включает биты для корректировки байт и выполняющие биты.

рабочий режим	WD (kbit/s)	новое предложение (kbit/s)	разница после перекодирования (kbit/s)	разница после перекодирования (% общего битрейта)
Test 1, 64s	64.00	62.78	-1.22	-1.90
Test 2, 32s	32.00	31.47	-0.53	-1.66
Test 3, 24s	24.00	23.61	-0.39	-1.64
Test 4, 20s	20.00	19.65	-0.35	-1.74
Test 5, 16s	16.00	15.72	-0.28	-1.75
Test 6, 24m	24.00	23.56	-0.44	-1.83
Test 7, 20m	20.00	19.61	-0.39	-1.95
Test 8, 16m	16.00	15.70	-0.30	-1.87
Test 9, 12m	12.00	11.77	-0.23	-1.92

ФИГ. 18

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

Минимальный и максимальный уровни битового резервуара арифметического кодера WD3 и предложенного кодера

рабочий режим	контроль битового резервуара					
	новое предложение			WD3		
	min	max	avg	min	max	avg
Test 1, 64kbps stereo	3739	9557	8874	2314	9557	7018
Test 2, 32kbps stereo	2335	4505	4293	582	4505	3529
Test 3, 24kbps stereo	2184	4704	4472	957	4704	3871
Test 4, 20kbps stereo	2688	4864	4660	712	4864	3854
Test 5, 16kbps stereo	2965	5006	4859	724	5006	4234
Test 6, 24kbps mono	2185	4704	4457	1002	4704	3927
Test 7, 20kbps mono	2782	4864	4690	1192	4864	3935
Test 8, 16kbps mono	2916	5006	4905	1434	5006	4450
Test 9, 12kbps mono	3645	5184	5107	2256	5184	4787

ФИГ. 19

Среднее значение сложности для декодирования битового потока 32 kbit/s WD3 для различных версий арифметического кодера

	WD3	Базовая версия
PCU (MHz)	0.953	0.823

ФИГ. 20

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```
static unsigned short ari_lookup_m[600] = {
0x02,0x01,0x03,0x38,0x3C,0x44,0x45,0x05,
0x07,0x37,0x41,0x08,0x0A,0x07,0x38,0x44,
0x0B,0x14,0x3E,0x3B,0x0C,0x14,0x3E,0x0C,
0x2B,0x5F,0x0F,0x42,0x3B,0x44,0x11,0x36,
0x42,0x41,0x13,0x2B,0x3E,0x3B,0x0C,0x14,
0x3E,0x14,0x2B,0x3E,0x15,0x07,0x3B,0x11,
0x25,0x42,0x41,0x16,0x36,0x3A,0x41,0x25,
0x36,0x3A,0x14,0x3E,0x36,0x42,0x36,0x3A,
0x2B,0x3E,0x42,0x38,0x41,0x17,0x19,0x42,
0x3B,0x44,0x19,0x1C,0x42,0x3B,0x44,0x1D,
0x2B,0x38,0x12,0x2B,0x1E,0x20,0x42,0x41,
0x22,0x36,0x37,0x41,0x24,0x36,0x42,0x41,
0x26,0x07,0x3A,0x2B,0x3E,0x01,0x3E,0x27,
0x07,0x37,0x44,0x3F,0x29,0x42,0x3B,0x44,
0x2A,0x07,0x37,0x41,0x29,0x07,0x38,0x2B,
0x3E,0x36,0x3E,0x39,0x42,0x34,0x07,0x3B,
0x26,0x36,0x42,0x41,0x36,0x42,0x3B,0x36,
0x3A,0x02,0x3A,0x03,0x38,0x07,0x37,0x07,
0x37,0x39,0x3A,0x3F,0x3B,0x41,0x44,0x57,
0x2C,0x07,0x3C,0x03,0x31,0x42,0x3C,0x22,
0x36,0x38,0x2A,0x07,0x27,0x2E,0x07,0x38,
0x44,0x2F,0x07,0x37,0x41,0x03,0x32,0x42,
0x3B,0x44,0x32,0x07,0x38,0x26,0x07,0x3A,
0x26,0x3E,0x02,0x2A,0x07,0x3B,0x33,0x03,
0x42,0x3B,0x44,0x1B,0x36,0x42,0x41,0x32,
0x07,0x3B,0x26,0x36,0x42,0x3B,0x36,0x3E,
0x39,0x42,0x39,0x42,0x3B,0x26,0x07,0x42,
0x41,0x36,0x42,0x37,0x41,0x36,0x42,0x3B,
0x07,0x3A,0x03,0x3B,0x07,0x42,0x3C,0x39,
0x42,0x3C,0x07,0x38,0x07,0x38,0x37,0x38,
0x3C,0x41,0x44,0x57,0x3B,0x3A,0x33,0x37,
0x33,0x03,0x02,0x33,0x07,0x3C,0x33,0x07,
0x3B,0x2A,0x34,0x42,0x41,0x34,0x07,0x38,
0x36,0x3E,0x26,0x07,0x3C,0x39,0x42,0x41,
0x33,0x36,0x42,0x3C,0x26,0x07,0x42,0x41,
0x07,0x38,0x07,0x3A,0x39,0x37,0x39,0x42,
0x3B,0x26,0x07,0x37,0x41,0x01,0x07,0x42,
0x3C,0x39,0x42,0x3B,0x03,0x3F,0x03,0x3B,
```

ФИГ. 21 (1)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

0x39,0x42,0x3B,0x39,0x37,0x3C,0x39,0x37,
0x3C,0x03,0x38,0x3A,0x3B,0x3C,0x41,0x44,
0x57,0x03,0x33,0x03,0x26,0x37,0x34,0x42,
0x41,0x39,0x3F,0x34,0x42,0x39,0x37,0x39,
0x42,0x3B,0x26,0x07,0x37,0x41,0x07,0x3B,
0x07,0x3A,0x03,0x42,0x41,0x39,0x42,0x3C,
0x39,0x42,0x3C,0x07,0x3F,0x03,0x3B,0x03,
0x3B,0x39,0x37,0x3C,0x02,0x42,0x3C,0x03,
0x3D,0x40,0x3C,0x41,0x44,0x57,0x03,0x39,
0x3D,0x03,0x3B,0x39,0x42,0x41,0x39,0x3A,
0x03,0x3F,0x39,0x3F,0x02,0x3E,0x3F,0x02,
0x40,0x3C,0x41,0x44,0x27,0x03,0x03,0x03,
0x3D,0x3F,0x40,0x39,0x41,0x44,0x03,0x3D,
0x42,0x43,0x03,0x3C,0x44,0x03,0x3D,0x40,
0x3C,0x02,0x3C,0x44,0x3D,0x43,0x3C,0x02,
0x41,0x44,0x3D,0x43,0x3C,0x03,0x44,0x3D,
0x43,0x41,0x02,0x44,0x3D,0x41,0x44,0x43,
0x41,0x44,0x43,0x41,0x44,0x03,0x3C,0x44,
0x46,0x47,0x49,0x4B,0x50,0x4D,0x4F,0x0B,
0x56,0x0C,0x0C,0x2B,0x03,0x52,0x0D,0x10,
0x1A,0x56,0x56,0x57,0x58,0x58,0x0C,0x26,
0x01,0x01,0x02,0x3D,0x1A,0x1E,0x56,0x5A,
0x5B,0x5B,0x00,0x27,0x15,0x0C,0x39,0x39,
0x26,0x01,0x02,0x02,0x3D,0x57,0x57,0x27,
0x57,0x00,0x39,0x39,0x02,0x02,0x03,0x27,
0x27,0x02,0x27,0x02,0x02,0x3D,0x5E,0x5D,
0x5F,0x5E,0x5D,0x5D,0x5D,0x5D,0x5C,0x5C,
0x5F,0x5D,0x5D,0x5D,0x5E,0x5D,0x5D,0x5C,
0x5C,0x5C,0x5C,0x5F,0x5D,0x5D,0x5D,0x5E,
0x5D,0x5C,0x5C,0x5E,0x5C,0x5E,0x5C,0x5C,
0x5F,0x5D,0x5C,0x5D,0x5F,0x5D,0x5D,0x5F,
0x5E,0x5F,0x5D,0x5F,0x5D,0x5F,0x5F,0x5D,
0x5F,0x5D,0x5F,0x5D,0x5F,0x5F,0x5F,
0x5F,0x5F,0x5F,0x5F,0x44,0x5F,0x5E,0x5D,
0x5D,0x5C,0x5C,0x5D,0x5C,0x5C,0x5C,0x5F,
0x5E,0x5D,0x5E,0x5D,0x5E,0x5D,0x5E,0x5F,
0x5E,0x5F,0x5E,0x5C,0x5E,0x5C,0x5E,0x5E,

```

```
};
```

ФИГ. 21 (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```
static unsigned long ari_hash_m[600] = {
0x00000100UL,0x00000302UL,0x0000053AUL,0x0000073BUL,0x00000A41UL,0x00000F44UL,
0x00111104UL,0x00111306UL,
0x00111542UL,0x0011173BUL,0x00111F44UL,0x00112209UL,0x0011242BUL,0x0011263EUL,
0x0011293CUL,0x00113105UL,
0x0011330CUL,0x0011352BUL,0x0011383AUL,0x00113F44UL,0x0011440CUL,0x0011462BUL,
0x00114F41UL,0x0011530CUL,
0x0012110DUL,0x0012120EUL,0x00121436UL,0x00121637UL,0x00121941UL,0x00122110UL,
0x00122312UL,0x00122507UL,
0x00122738UL,0x00123156UL,0x00123314UL,0x00123507UL,0x0012373AUL,0x00123F44UL,
0x00124212UL,0x0012452BUL,
0x00124F44UL,0x00125314UL,0x0012762BUL,0x00131121UL,0x00131323UL,0x00131542UL,
0x0013211DUL,0x00132216UL,
0x00132436UL,0x00132738UL,0x0013311DUL,0x00133329UL,0x00133507UL,0x00133838UL,
0x00134115UL,0x0013432BUL,
0x0013463EUL,0x00134F44UL,0x0013532BUL,0x0013FF3BUL,0x00142307UL,0x00143126UL,
0x00143407UL,0x00143E44UL,
0x00144307UL,0x0014FF3BUL,0x0015633EUL,0x0017863BUL,0x0021005AUL,0x00211218UL,
0x00211407UL,0x00211637UL,
0x00211941UL,0x0021211AUL,0x0021221BUL,0x00212436UL,0x00212637UL,0x00212941UL,
0x00213118UL,0x00213320UL,
0x00213507UL,0x00213F44UL,0x00214314UL,0x00220001UL,0x0022121FUL,0x00221407UL,
0x0022173BUL,0x00222121UL,
0x00222323UL,0x00222542UL,0x0022283BUL,0x0022311DUL,0x00223325UL,0x00223507UL,
0x00223737UL,0x00224115UL,
0x00224436UL,0x0022463EUL,0x00224F44UL,0x00225436UL,0x00225F44UL,0x0022622BUL,
0x0023112DUL,0x00231330UL,
0x00231542UL,0x0023183CUL,0x0023212DUL,0x00232228UL,0x00232407UL,0x00232637UL,
0x00232941UL,0x00233115UL,
0x0023332BUL,0x00233542UL,0x0023383BUL,0x0023412AUL,0x00234336UL,0x00234642UL,
0x00234F44UL,0x00235407UL,
0x00235F44UL,0x0023653EUL,0x0023FF3BUL,0x00241307UL,0x00241F44UL,0x00242336UL,
0x00242542UL,0x00242F44UL,
0x00243234UL,0x00243407UL,0x00243738UL,0x00244126UL,0x00244307UL,0x0024463AUL,
0x00244F44UL,0x00245442UL,
0x00245F44UL,0x00246236UL,0x0024FF3CUL,0x00252442UL,0x00252F44UL,0x00253303UL,
0x00253F44UL,0x00254303UL,
0x00254F44UL,0x00255303UL,0x0025FF41UL,0x0026733EUL,0x0027FF44UL,0x002AF203UL,
0x002FFF44UL,0x0031115BUL,
0x00311331UL,0x00311542UL,0x00312121UL,0x0031222DUL,0x00312436UL,0x00312637UL,
0x00312F44UL,0x00313335UL,
```

ФИГ. 22 (2)



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

0x00313507UL,0x00313F44UL,0x00314332UL,0x0031FF3CUL,0x0032112CUL,0x00321335UL,  
0x00321542UL,0x0032183CUL,  
0x0032212CUL,0x00322330UL,0x00322542UL,0x0032273BUL,0x0032312DUL,0x00323231UL,  
0x00323407UL,0x00323637UL,  
0x00323A41UL,0x0032412AUL,0x00324407UL,0x00324642UL,0x00324F44UL,0x00325336UL,  
0x00325507UL,0x00325F44UL,  
0x00326234UL,0x0032FF3CUL,0x00331127UL,0x00331332UL,0x00331542UL,0x0033212EUL,  
0x00332334UL,0x00332407UL,  
0x00332637UL,0x00332941UL,0x00333115UL,0x00333235UL,0x00333407UL,0x00333738UL,  
0x0033412AUL,0x00334336UL,  
0x00334637UL,0x00334F44UL,0x00335234UL,0x00335407UL,0x0033573AUL,0x00335F44UL,  
0x00336407UL,0x0033FF3CUL,  
0x00341307UL,0x00341F44UL,0x00342307UL,0x00342542UL,0x00342F44UL,0x00343234UL,  
0x00343442UL,0x0034373BUL,  
0x00344126UL,0x00344307UL,0x00344542UL,0x0034483BUL,0x00345126UL,0x00345307UL,  
0x00345637UL,0x00345F44UL,  
0x00346442UL,0x0034FF3CUL,0x00352442UL,0x00353139UL,0x00353303UL,0x00353637UL,  
0x00353F44UL,0x00354303UL,  
0x00354637UL,0x00354F44UL,0x00355442UL,0x00356102UL,0x00356303UL,0x0035FF41UL,  
0x00366203UL,0x0037733DUL,  
0x0039E43AUL,0x003BF641UL,0x003FFF44UL,0x00411227UL,0x00411334UL,0x0041212CUL,  
0x00412407UL,0x0041312EUL,  
0x00413334UL,0x0041FF3CUL,0x00421127UL,0x00421334UL,0x00421542UL,0x00422127UL,  
0x00422334UL,0x00422542UL,  
0x00422F44UL,0x00423232UL,0x00423407UL,0x0042373BUL,0x00424126UL,0x00424336UL,  
0x00424542UL,0x00424F44UL,  
0x00425407UL,0x0042FF3CUL,0x00431339UL,0x00431542UL,0x00432133UL,0x00432407UL,  
0x0043273BUL,0x00432F44UL,  
0x00433234UL,0x00433407UL,0x00433637UL,0x00433F44UL,0x00434234UL,0x00434442UL,  
0x00434738UL,0x00435126UL,  
0x00435542UL,0x00435F44UL,0x00436442UL,0x0043FF41UL,0x00441303UL,0x00442126UL,  
0x00442307UL,0x00442542UL,  
0x00442F44UL,0x00443239UL,0x00443442UL,0x0044373BUL,0x00443F44UL,0x00444234UL,  
0x00444442UL,0x00444637UL,  
0x00444F44UL,0x00445307UL,0x00445637UL,0x00445F44UL,0x00446537UL,0x0044FF41UL,  
0x00452442UL,0x00452F44UL,  
0x00453303UL,0x00453537UL,0x00453F44UL,0x00454303UL,0x00454638UL,0x00454F44UL,  
0x00455303UL,0x00455638UL,  
0x00455F44UL,0x00456437UL,0x0045FF41UL,0x00466203UL,0x00478438UL,0x0049FF44UL,  
0x004CF741UL,0x004FFF44UL,

ФИГ. 22 (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

0x00511226UL,0x00512127UL,0x00512339UL,0x00521127UL,0x00521339UL,0x00522127UL,  
 0x00522339UL,0x00522637UL,  
 0x00523126UL,0x00523403UL,0x00524126UL,0x00524307UL,0x00531126UL,0x00531307UL,  
 0x00532126UL,0x00532307UL,  
 0x00532537UL,0x00532F44UL,0x00533239UL,0x00533442UL,0x0053373BUL,0x00534126UL,  
 0x00534542UL,0x00534F44UL,  
 0x00535442UL,0x0053FF3CUL,0x00542303UL,0x00542638UL,0x00543102UL,0x00543307UL,  
 0x00543638UL,0x00543F44UL,  
 0x00544307UL,0x00544537UL,0x00545102UL,0x00545303UL,0x0054FF41UL,0x00552437UL,  
 0x00552F44UL,0x00553437UL,  
 0x00553F44UL,0x00554303UL,0x0055463BUL,0x00554F44UL,0x00555307UL,0x00555537UL,  
 0x00556102UL,0x00556342UL,  
 0x00566203UL,0x00577342UL,0x0059733AUL,0x005CF53CUL,0x005FFF44UL,0x00611239UL,  
 0x00621127UL,0x00623307UL,  
 0x00631126UL,0x00632442UL,0x00632F44UL,0x00633303UL,0x00633638UL,0x00634102UL,  
 0x00634303UL,0x0063FF41UL,  
 0x00642303UL,0x00642F44UL,0x00643303UL,0x00643F44UL,0x00644207UL,0x00655203UL,  
 0x00665F44UL,0x00666303UL,  
 0x00677203UL,0x0069A53BUL,0x006DF841UL,0x006FFF44UL,0x00711239UL,0x00721127UL,  
 0x00722239UL,0x00733239UL,  
 0x00744437UL,0x00755203UL,0x00776F44UL,0x00777437UL,0x007BF33FUL,0x007FFF44UL,  
 0x00822239UL,0x00833203UL,  
 0x00854540UL,0x00888102UL,0x00888538UL,0x008BF23DUL,0x008FFF44UL,0x00922239UL,  
 0x0094333DUL,0x0096533DUL,  
 0x00998F44UL,0x00999303UL,0x009BF53BUL,0x009FFF44UL,0x00A44203UL,0x00A6633FUL,  
 0x00AA9F44UL,0x00AAA303UL,  
 0x00ADF33DUL,0x00AFF44UL,0x00B33203UL,0x00B55440UL,0x00BBAC44UL,0x00BBB53BUL,  
 0x00BFFF44UL,0x00C33203UL,  
 0x00C6423DUL,0x00CCBF44UL,0x00CCC303UL,0x00CFFF44UL,0x00D4423DUL,0x00DDD303UL,  
 0x00DFFF44UL,0x00E6453CUL,  
 0x00EEE342UL,0x00EFFF44UL,0x00F55343UL,0x00F7FF44UL,0x00FFEF44UL,0x00FFF33DUL,  
 0x00FFF744UL,0x01000145UL,  
 0x01011147UL,0x01111248UL,0x0111224AUL,0x0111324DUL,0x0112114CUL,0x0112214EUL,  
 0x01123108UL,0x01131150UL,  
 0x01132150UL,0x01133150UL,0x01141126UL,0x01144100UL,0x01211151UL,0x01212153UL,  
 0x01213108UL,0x01221154UL,  
 0x01222155UL,0x01223117UL,0x01224212UL,0x01231215UL,0x01232215UL,0x01233215UL,  
 0x01241126UL,0x01242239UL,  
 0x0124322BUL,0x01251103UL,0x01255100UL,0x01311159UL,0x01312155UL,0x01313117UL,  
 0x01315201UL,0x0132122CUL,

ФИГ. 22 (3)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

0x0132222CUL,0x0132321DUL,0x01331157UL,0x01332158UL,0x01333150UL,0x01341126UL,
0x01342127UL,0x01343100UL,
0x01344100UL,0x01351103UL,0x01355100UL,0x01369100UL,0x0141115AUL,0x01421227UL,
0x01422227UL,0x01431226UL,
0x01432226UL,0x01441127UL,0x01442127UL,0x01443100UL,0x01444100UL,0x01458100UL,
0x01511157UL,0x01531239UL,
0x01555100UL,0x01611157UL,0x01631239UL,0x01777100UL,0x01D33102UL,0x01FFF203UL,
0x0200055DUL,0x0200085DUL,
0x02000F5FUL,0x0211155DUL,0x02111F44UL,0x02112F5FUL,0x02121F44UL,0x02122F44UL,
0x02133F5FUL,0x0217FF5FUL,
0x021FFF44UL,0x02211F44UL,0x02212F44UL,0x02221F44UL,0x0222245EUL,0x02222F5FUL,
0x02223F5FUL,0x02232F5FUL,
0x02233F5FUL,0x02243F5FUL,0x022BFF5FUL,0x022FFF44UL,0x02322F44UL,0x02323F5FUL,
0x02332F5FUL,0x0233355CUL,
0x02333F5FUL,0x02334F5FUL,0x0233FF5CUL,0x02343F5FUL,0x0234FF5CUL,0x02353F5FUL,
0x02364F5FUL,0x0239655CUL,
0x023FFF44UL,0x02433F5FUL,0x0246445CUL,0x024A955DUL,0x024FFF44UL,0x0257435EUL,
0x025AC55CUL,0x025FFF44UL,
0x0267565DUL,0x026FFF44UL,0x0278655DUL,0x027FFF44UL,0x0288875DUL,0x028CC95DUL,
0x028FFF44UL,0x029A965DUL,
0x029FFF44UL,0x02ABA65DUL,0x02AFF44UL,0x02BAFF5FUL,0x02BFFF44UL,0x02CCC45DUL,
0x02CFFF44UL,0x02DFFF44UL,
0x02EEE65DUL,0x02EFFF44UL,0x02F7335EUL,0x02FF0044UL,0x02FFEF44UL,0x02FFFF44UL,
0x0400045EUL,0x04000F5FUL,
0x04111F5DUL,0x04234F5DUL,0x042DFF5CUL,0x04343F5DUL,0x043FFF5FUL,0x044FFF5FUL,
0x045ED75CUL,0x045FFF5FUL,
0x046DFF5FUL,0x046FFF5FUL,0x047B5C5CUL,0x047FFF5FUL,0x048B435EUL,0x048FFF5FUL,
0x0499FF5CUL,0x04EFFF5FUL,
0x04FDFE5DUL,0x04FFFF5FUL,0x0600075EUL,0x06DFFF5FUL,0x06FFFF5FUL,0x08BFFF5CUL,
0x08FFFF5CUL,0x7FFFFFFF4FUL,

```

```
};
```

ФИГ. 22 (4)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

unsigned short ari_cf_m [96][17] = {
  { 7529, 5263, 5173, 5162, 2636, 825, 674, 653, 511, 281, 210, 195, 173, 130, 105,
    96,
    0
  },
  { 10351, 7392, 7203, 7178, 4327, 1620, 1279, 1230, 1008, 606, 436, 399, 362, 288, 233,
    212,
    0
  },
  { 12505, 9566, 9216, 9157, 6631, 3220, 2541, 2418, 2086, 1404, 1024, 922, 858, 721,
    597, 535,
    0
  },
  { 14710, 12600, 11956, 11801, 10114, 7056, 5723, 5381, 4870, 3724, 2870, 2566, 2437, 2122,
    1809, 1609,
    0
  },
  { 4186, 2623, 2608, 2607, 939, 109, 86, 84, 63, 23, 14, 13, 11, 8, 5, 4,
    0
  },
  { 7310, 4598, 4547, 4544, 2079, 354, 264, 259, 204, 90, 42, 38, 34, 25, 14, 11,
    0
  },
  { 9998, 6785, 6641, 6630, 4087, 1058, 770, 746, 621, 339, 166, 143, 131, 104, 67,
    49,
    0
  },
  { 13801, 11125, 10550, 10472, 8498, 5030, 3694, 3509, 3101, 2141, 1247, 1038, 975, 821,
    623, 440,
    0
  },
  { 5784, 3557, 3523, 3521, 1359, 164, 122, 119, 85, 29, 16, 14, 12, 8, 5, 4,
    0
  },
  { 7617, 4716, 4649, 4645, 2129, 361, 258, 252, 191, 80, 39, 34, 30, 22, 13, 10,
    0
  },
  { 9553, 6223, 6064, 6052, 3524, 912, 643, 620, 501, 262, 132, 112, 102, 79, 52, 39,
    0
  },
  { 8423, 5300, 5186, 5179, 2629, 539, 375, 362, 276, 119, 61, 53, 46, 33, 21, 17,
    0
  },
  { 9570, 6162, 5952, 5936, 3574, 1016, 702, 670, 538, 282, 152, 129, 117, 90, 63,
    50,
    0
  },
  },
};

```

2310 →

2312 →

ФИГ. 23 (1)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 7355, 4678, 4630, 4628, 1834, 252, 190, 187, 124, 38, 21, 19, 16, 10, 6, 5,
0
},
{ 8916, 5717, 5627, 5622, 2647, 471, 340, 332, 242, 97, 46, 41, 36, 25, 15, 12,
0
},
{ 10839, 7394, 7198, 7184, 4273, 1169, 832, 803, 643, 327, 167, 143, 130, 100, 66,
49,
0
},
{ 8036, 5132, 5045, 5040, 2207, 350, 250, 243, 167, 56, 31, 28, 23, 15, 10, 8,
0
},
{ 9663, 6256, 6110, 6101, 3064, 633, 441, 426, 311, 127, 65, 57, 49, 34, 22, 18,
0
},
{ 11237, 7629, 7361, 7339, 4460, 1302, 898, 859, 675, 333, 178, 152, 136, 104, 73,
57,
0
},
{ 10436, 6930, 6718, 6703, 3735, 902, 619, 595, 446, 189, 98, 85, 74, 51, 34, 28,
0
},
{ 11678, 7933, 7574, 7541, 4826, 1560, 1057, 1002, 796, 406, 218, 184, 165, 124, 88,
70,
0
},
{ 10597, 7265, 7057, 7039, 3904, 1071, 768, 738, 541, 227, 127, 111, 93, 60, 40,
34,
0
},
{ 11298, 7892, 7596, 7566, 4416, 1276, 873, 832, 611, 261, 141, 122, 103, 70, 47,
40,
0
},
{ 6435, 4238, 4210, 4208, 1558, 216, 174, 172, 116, 39, 24, 22, 18, 11, 7, 6,
0
},
{ 8744, 5744, 5671, 5667, 2636, 496, 375, 368, 273, 113, 58, 52, 45, 32, 19, 15,
0
},
{ 10899, 7632, 7457, 7444, 4488, 1270, 942, 914, 737, 383, 204, 177, 161, 122, 80,
62,
0
},
},

```

ФИГ. 23 (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 7929, 5176, 5119, 5116, 2169, 307, 230, 226, 155, 51, 28, 25, 21, 13, 8, 7,
  0
},
{ 9520, 6304, 6204, 6199, 3044, 585, 424, 415, 304, 118, 59, 52, 45, 31, 19, 15,
  0
},
{ 11187, 7805, 7600, 7586, 4554, 1292, 928, 896, 703, 342, 178, 153, 138, 103, 68,
  52,
  0
},
{ 10371, 7037, 6881, 6871, 3721, 860, 613, 596, 440, 176, 88, 77, 67, 46, 28, 23,
  0
},
{ 8562, 5804, 5738, 5734, 2472, 467, 374, 369, 233, 76, 48, 44, 34, 20, 13, 11,
  0
},
{ 10078, 6838, 6713, 6705, 3407, 782, 591, 578, 407, 164, 92, 83, 69, 46, 29, 24,
  0
},
{ 11694, 8342, 8108, 8089, 4999, 1559, 1143, 1105, 858, 424, 240, 210, 185, 135, 92,
  73,
  0
},
{ 9115, 6191, 6084, 6077, 2924, 604, 455, 445, 304, 109, 64, 58, 47, 29, 20, 17,
  0
},
{ 10786, 7434, 7256, 7244, 3968, 1006, 734, 712, 517, 213, 118, 104, 88, 60, 40,
  34,
  0
},
{ 12216, 8840, 8535, 8507, 5492, 1860, 1342, 1289, 1007, 505, 289, 249, 221, 163, 115,
  94,
  0
},
{ 11473, 8085, 7843, 7825, 4602, 1304, 931, 898, 669, 283, 153, 135, 116, 80, 53,
  44,
  0
},
{ 12650, 9279, 8908, 8872, 5924, 2160, 1539, 1470, 1162, 585, 329, 282, 251, 184, 130,
  107,
  0
},
{ 12597, 9310, 8859, 8806, 6019, 2457, 1741, 1643, 1310, 684, 401, 343, 304, 225, 164,
  137,
  0
},
},

```

ФИГ. 23 (3)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 11509, 8255, 8030, 8012, 4706, 1489, 1135, 1103, 793, 347, 212, 192, 154, 98, 68,
59,
0
},
{ 11946, 8648, 8370, 8344, 5106, 1628, 1185, 1142, 842, 366, 206, 183, 153, 100, 68,
57,
0
},
{ 13088, 9856, 9444, 9397, 6468, 2535, 1831, 1745, 1384, 700, 407, 352, 310, 224, 160,
132,
0
},
{ 12262, 9021, 8683, 8647, 5519, 1929, 1405, 1342, 1024, 481, 280, 245, 210, 146, 104,
89,
0
},
{ 13353, 10225, 9742, 9678, 6908, 2945, 2133, 2017, 1619, 875, 524, 450, 401, 298, 218,
183,
0
},
{ 10055, 6990, 6879, 6872, 3544, 882, 687, 675, 490, 209, 121, 112, 95, 63, 41, 36,
0
},
{ 10647, 7427, 7278, 7270, 3911, 968, 723, 705, 513, 203, 109, 98, 83, 55, 34, 28,
0
},
{ 11221, 8027, 7861, 7851, 4397, 1264, 981, 961, 689, 293, 174, 159, 132, 83, 54,
46,
0
},
{ 11700, 8429, 8209, 8192, 4825, 1451, 1079, 1049, 767, 327, 186, 167, 140, 92, 61,
52,
0
},
{ 12949, 9762, 9414, 9379, 6351, 2418, 1786, 1717, 1351, 686, 398, 345, 301, 216, 152,
125,
0
},
{ 12208, 8979, 8703, 8682, 5437, 1780, 1303, 1261, 955, 422, 231, 204, 175, 120, 77,
63,
0
},
{ 13277, 10148, 9741, 9698, 6807, 2800, 2047, 1960, 1563, 808, 463, 398, 352, 255, 174,
140,
0
},
},

```

ФИГ. 23 (4)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 12426, 9284, 8976, 8947, 5753, 2124, 1609, 1555, 1166, 548, 333, 298, 247, 162, 113,
99,
0
},
{ 13492, 10473, 10042, 9988, 7177, 3133, 2347, 2244, 1797, 970, 594, 520, 457, 331,
241, 204,
0
},
{ 12407, 9334, 9013, 8977, 5920, 2276, 1708, 1642, 1284, 642, 392, 348, 299, 211, 153,
132,
0
},
{ 13579, 10631, 10182, 10121, 7466, 3396, 2542, 2420, 1975, 1112, 691, 604, 542, 408,
300, 256,
0
},
{ 15415, 14163, 13591, 13335, 12111, 9885, 8640, 8070, 7541, 6363, 5413, 4919, 4688, 4157,
3662, 3364,
0
},
{ 15645, 14640, 14124, 13838, 12910, 11112, 9969, 9338, 8909, 7923, 7060, 6507, 6315, 5841,
5365, 5003,
0
},
{ 13803, 11130, 10691, 10636, 7807, 3900, 3070, 2956, 2343, 1300, 866, 781, 667, 462,
337, 292,
0
},
{ 15381, 14112, 13568, 13348, 12083, 9783, 8595, 8105, 7565, 6347, 5408, 4945, 4704, 4158,
3671, 3377,
0
},
{ 15595, 14555, 14062, 13813, 12830, 10944, 9820, 9259, 8813, 7769, 6872, 6351, 6149, 5655,
5160, 4798,
0
},
{ 15855, 15124, 14719, 14487, 13812, 12440, 11517, 10975, 10628, 9802, 9036, 8493, 8312,
7869, 7383, 6992,
0
},
{ 15484, 14324, 13800, 13578, 12460, 10325, 9149, 8632, 8118, 6954, 6030, 5548, 5320, 4797,
4296, 3961,
0
},
{ 15407, 14130, 13600, 13402, 12107, 9730, 8527, 8059, 7475, 6205, 5245, 4803, 4563, 4022,
3557, 3260,
0
},
}

```

ФИГ. 23 (5)



Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 15475,14322,13830,13621,12482,10384, 9270, 8779, 8266, 7125, 6200, 5730, 5493, 4962,
4469, 4140,
0
},
{ 15596,14601,14153,13948,12948,11040, 9997, 9517, 9048, 7949, 7054, 6560, 6325, 5785,
5280, 4913,
0
},
{ 15871,15158,14789,14602,13921,12540,11660,11198,10847,10009, 9224, 8719, 8532,
8078, 7594, 7190,
0
},
{ 15474,14312,13843,13669,12397,10171, 9109, 8697, 8063, 6764, 5886, 5470, 5181, 4574,
4090, 3790,
0
},
{ 15720,14858,14443,14249,13373,11680,10700,10226, 9779, 8747, 7881, 7386, 7156,
6611, 6097, 5712,
0
},
{ 16133,15786,15587,15472,15104,14369,13854,13555,13314,12738,12214,11861,11701,
11314,10905,10579,
0
},
{ 2594, 1645, 1633, 1631, 535, 108, 92, 89, 64, 34, 27, 25, 21, 15, 12, 11,
0
},
{ 7130, 4521, 4444, 4431, 2088, 598, 489, 469, 378, 235, 185, 173, 153, 120, 100,
94,
0
},
{ 1326, 780, 778, 777, 173, 17, 15, 14, 11, 7, 6, 5, 4, 3, 2, 1,
0
},
{ 4974, 2832, 2814, 2812, 945, 102, 78, 76, 56, 26, 16, 15, 13, 9, 6, 5,
0
},
{ 3593, 2051, 2043, 2042, 530, 36, 28, 27, 18, 8, 6, 5, 4, 3, 2, 1,
0
},
{ 5521, 3055, 3028, 3026, 1052, 102, 72, 69, 49, 20, 12, 11, 9, 6, 4, 3,
0
},
{ 4294, 2539, 2520, 2518, 812, 83, 61, 59, 41, 17, 12, 11, 9, 6, 4, 3,
0
},
},

```

ФИГ. 23 (6)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

{ 4456, 2667, 2653, 2652, 724, 67, 54, 53, 31, 11, 8, 7, 5, 3, 2, 1,
0
},
{ 6684, 3922, 3872, 3869, 1425, 186, 133, 129, 86, 35, 21, 19, 15, 9, 6, 5,
0
},
{ 5087, 3024, 2988, 2985, 965, 99, 71, 69, 43, 15, 10, 9, 7, 4, 3, 2,
0
},
{ 7587, 4403, 4297, 4289, 1806, 279, 181, 171, 118, 49, 30, 27, 22, 15, 11, 10,
0
},
{ 6242, 4036, 3962, 3957, 1604, 337, 256, 251, 154, 49, 31, 28, 20, 10, 6, 5,
0
},
{ 3727, 2352, 2346, 2345, 568, 53, 46, 45, 27, 9, 7, 6, 5, 3, 2, 1,
0
},
{ 6369, 3904, 3876, 3874, 1372, 186, 150, 147, 101, 42, 27, 25, 20, 13, 9, 8,
0
},
{ 5012, 3060, 3046, 3045, 921, 76, 60, 59, 37, 13, 9, 8, 6, 4, 3, 2,
0
},
{ 5471, 3590, 3569, 3568, 1118, 176, 153, 151, 83, 26, 19, 18, 13, 7, 5, 4,
0
},
{ 5866, 3774, 3736, 3733, 1366, 204, 163, 159, 101, 37, 26, 24, 19, 11, 8, 7,
0
},
{ 8648, 5538, 5437, 5427, 2527, 521, 397, 382, 269, 121, 81, 75, 60, 40, 30, 27,
0
},
{ 7300, 5124, 5049, 5043, 2284, 682, 582, 574, 341, 124, 90, 86, 59, 29, 20, 18,
0
},
{ 7183, 4913, 4817, 4808, 2167, 517, 404, 393, 252, 92, 63, 59, 43, 24, 17, 15,
0
},
{ 4749, 3285, 3273, 3272, 938, 157, 141, 140, 81, 27, 21, 20, 15, 8, 6, 5,
0
},
{ 6602, 4705, 4676, 4674, 1689, 388, 347, 344, 192, 63, 49, 47, 33, 16, 11, 10,
0
},
},

```

ФИГ. 23 (7)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая модификацию числового представления числового значения предыдущего контекста

```

    { 6849, 4648, 4599, 4596, 1868, 367, 304, 299, 187, 68, 49, 46, 34, 19, 14, 13,
      0
    },
    { 16383, 16382, 14098, 13672, 13671, 13670, 11058, 10499, 8080, 5456, 4230, 3829, 3412,
      2876, 2494, 2264,
      0
    },
    { 16383, 16382, 14436, 14062, 14061, 14060, 11574, 11038, 8597, 5905, 4621, 4204, 3720,
      3106, 2674, 2430,
      0
    },
    { 16383, 16382, 14635, 14264, 14263, 14262, 11898, 11372, 8910, 6216, 4935, 4506, 3962,
      3287, 2835, 2578,
      0
    },
    { 16383, 16382, 14851, 14380, 14379, 14378, 12610, 12032, 9670, 7609, 6543, 6085, 5335,
      4590, 4112, 3825,
      0
    }
  };

```

2396 →

ФИГ. 23 (8)

```
unsigned short ari_cf_r [4] = {(3 << 14)/4, (2 << 14)/4, (1 << 14)/4, 0};
```

ФИГ. 24